Examining the geometry of neural mode connecting loss subspaces

Ting Chen
Department of Computer Science
Columbia University
UNI:zc2666

Tristan Luca Saidi
Department of Computer Science
Columbia University
UNI:tls2160

Abstract

Recent works in exploring high dimensional neural network loss landscapes have expanded our understanding of these non-convex optimization spaces. In particular, it has been discovered that seemingly disparate local minima are connected by low-loss, high-accuracy curves and simplexes, a phenomenon known as mode connectivity. In this work we look to analyze some of the geometric properties of these low loss subspaces, including intrinsic dimensionality, curvature, and sensitivity to perturbations. We perform these analyses across three classes of subspaces: lines parameterized by two endpoints (1-simplexes), triangles parameterized by 3 endpoints (2-simplexes), and finally nonlinear curves parameterized by a neural network. Overall we verify that subspace fitting provides an effective way to visualize and sample from low loss regions of weight space. We also extract information about the nonlinear structure of the space that will inform future extensions of this project to higher dimensional nonlinear subspace learning and geometry analysis.

1 Introduction

Historically, multi-layer neural network loss landscapes have been thought of as high dimensional and extremely nonconvex surfaces, where stochastic gradient descent (SGD) training from different random initializations converge to different isolated local minima throughout the surface. Garipov et al. [2018], Draxler et al. [2018], Benton et al. [2021] discovered that those modes found through SGD training of randomly initialized networks are in fact connected through low loss, high accuracy curves and simplexes in parameter space.

An interesting phenomena in and of itself, mode connectivity also has important applications for practitioners. Other recent works aimed at characterizing the implicit biases of SGD have shown that SGD is predisposed towards low curvature regions of the space, and in fact converges to local minima due to an in-built negative feedback loop that continuously keeps the optimization trajectory in low-curvature regions during training [Damian et al., 2023, Gilmer et al., 2022] Mode connectivity can provide a way to access these low-curvature regions for analysis and efficient model combination (ensembling methods).

In this work we aim for the former as we perform intrinsic dimensionality estimation, curvature estimation, and perturbation analysis on various classes of low loss subspaces. These analyses are done on a one hidden layer multilayer perceptron (MLP) trained on the Fashion-MNIST dataset [Xiao et al., 2017]. [1]

[1] Code is available here: https://github.com/tingtang2/loss-subspace-geometry

Preprint. Under review.

2 Related Works

Garipov et al. [2018], Draxler et al. [2018] were the first works to report on and characterize mode connectivity as lines and curves connecting the modes of independently trained networks. Shortly following, Fort and Jastrzebski [2019] also reported on the behavior by characterizing it in the context of their model of the loss surface. This model consists of a union of high dimensional manifolds which they term as *wedges*. Benton et al. [2021], Wortsman et al. [2021] subsequently expanded the concept of mode connectivity to include simplexes and simplicial complexes as parameterizations of the low loss subspaces. Crucially Wortsman et al. [2021] developed an algorithm to find learn these low loss subspaces directly through training, which we discuss in 3.1.

Also relevant and part of the inspiration for this work is Skorokhodov and Burtsev [2019], who extended the method from Garipov et al. [2018] to find planes that resembled any given 2 dimensional image. This demonstrated the complexness of neural network loss landscapes and motivated us to further explore finding useful structures in this space.

3 Methods

3.1 Subspace finding

Throughout our experiments we analyze three different parameterizations of loss subspaces for the neural network $f(\mathbf{x};\theta)$:

- Lines parameterized by two endpoints (1-simplexes), denoted by $\mathtt{P}(\alpha;\boldsymbol{\omega}_1,\boldsymbol{\omega}_2) = (1-\alpha)\boldsymbol{\omega}_1 + \alpha\boldsymbol{\omega}_2$, where $\alpha \in [0,1]$
- Triangles parameterized by 3 endpoints (2-simplexes), denoted by $\mathtt{P}(\alpha_1,\alpha_2;\boldsymbol{\omega}_1,\boldsymbol{\omega}_2,\boldsymbol{\omega}_3) = (1-\alpha_1-\alpha_2)\boldsymbol{\omega}_1 + \alpha_1\boldsymbol{\omega}_2 + \alpha_2\boldsymbol{\omega}_3$, where $\alpha_1 + \alpha_2 \in [0,1]$
- Nonlinear curves parameterized by a neural network, denoted by $\mathtt{g}(\alpha;\phi)$, where $\alpha \in [0,1]$, $\mathtt{g} : \mathbb{R}^1 \to \mathbb{R}^{\boldsymbol{\omega}}$ is a neural network, ϕ are the parameters of the neural network, and $\boldsymbol{\omega}$ is the size of the parameter space of θ.

3.1.1 Affine Subspaces

Throughout the rest of this work, we use the subspace finding method from Wortsman et al. [2021], presented in Algorithm 1. For our work we consider learning lines and triangles (1- and 2- simplexes) for convenience, however this method generalizes to higher dimensions and other functional forms. In fact, we extend this algorithm to find loss subspaces parameterized by a non-linear function in the form of a MLP, as described in the next section.

Algorithm 1 Affine Subspace finding algorithm from Wortsman et al. [2021]

Input: Function defining line in weight space $\mathtt{P}$ with endpoints $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2$, network function f, training set $\mathcal{S}$, loss function l, regularization parameter β

```
Independently initialize $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2$.
for batch $(\mathbf{x},\mathbf{y}) \in \mathcal{S}$ do
    Sample $\alpha$ uniformly from $[0,1]$.    ▷ $\alpha$ defines where on the line you sample weights from
    $\boldsymbol{\theta} \leftarrow \mathtt{P}(\alpha;\boldsymbol{\omega}_1,\boldsymbol{\omega}_2)$
    $\hat{\mathbf{y}} \leftarrow f(\mathbf{x};\boldsymbol{\theta})$
    $\mathcal{L} \leftarrow l(\hat{\mathbf{y}},\mathbf{y}) + \beta\cos^2(\boldsymbol{\omega}_1,\boldsymbol{\omega}_2)$    ▷ regularization to encourage diversity between $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2$
    Backprop from $\mathcal{L}$, update each $\{\boldsymbol{\omega}_i\}_{i=1}^2$ with $\frac{\partial\mathcal{L}}{\partial\boldsymbol{\omega}_i} = \frac{\partial l}{\partial\boldsymbol{\theta}}\frac{\partial\mathtt{P}}{\partial\boldsymbol{\omega}_i} + \beta\frac{\partial\cos^2(\boldsymbol{\omega}_1,\boldsymbol{\omega}_2)}{\partial\boldsymbol{\omega}_i}$.
end for
```

Formally, consider $\boldsymbol{\omega}_1 \in \mathbb{R}^N$ and $\boldsymbol{\omega}_2 \in \mathbb{R}^N$ as the endpoints of a line in weight space defined by $\mathtt{P}(\alpha;\boldsymbol{\omega}_1,\boldsymbol{\omega}_2) = (1-\alpha)\boldsymbol{\omega}_1 + \alpha\boldsymbol{\omega}_2$, where $\alpha \in [0,1]$. The training procedure above seeks to optimize parameters $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2$ such that the line defined by $\mathtt{P}(\alpha;\boldsymbol{\omega}_1,\boldsymbol{\omega}_2)$ contains solutions with high test accuracy for all values of α. This is achieved by optimizing the training objective

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[\mathbb{E}_{\alpha\sim\text{Uniform}(0,1)}[l(f(\mathbf{x},\mathtt{P}(\alpha;\boldsymbol{\omega}_1,\boldsymbol{\omega}_2)),\mathbf{y})]] \tag{1}$$

where $\mathcal{D}$ denotes the data distribution, f denotes the function mapping between $\mathbf{x}$ and $\mathbf{y}$, and l denotes a loss function.

Practically, we optimize Equation 1 through stochastic updates and backpropagation. Inspired by Fort et al. [2019], the method above also includes a regularization term in order to explicitly encourage functional diversity along the line parameterized by $\mathtt{P}(\alpha; \boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$. This is done by encouraging $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ to have a cosine similarity 0. Overall, this method differs subtly from popular methods such as those found in Garipov et al. [2018] and Benton et al. [2021], as those methods begin with randomly initialized networks that have already been trained a priori before proceeding to find the connecting low loss subspace. This method learns the loss subspace directly during training by keeping track of the functional form of each kind subspace and updating its parameters.

3.1.2 Nonlinear Subspaces

As mentioned, we also extend the subspace finding method from Wortsman et al. [2021] to the nonlinear case by parameterizing our subspace with a MLP. Our optimization procedure is analogous to Algorithm 1, where gradients from the network f backpropagate to our non-linear parameterized subspace.

Algorithm 2 Nonlinear Subspace finding algorithm

```
Input: Function g parameterized by $\phi$ that maps scalar $\alpha$ to weight space, network function $f$, training set $\mathcal{S}$, loss function $l$, regularization parameter $\beta$
Independently initialize $\theta$, $\phi$.
for batch $(\mathbf{x}, \mathbf{y}) \in \mathcal{S}$ do
    Sample $\alpha$ uniformly from $[0, 1]$.    ▷ $\alpha$ defines where on the subspace you sample weights from
    $\boldsymbol{\theta} \leftarrow \mathrm{g}(\alpha; \phi)$
    $\hat{\mathbf{y}} \leftarrow f(\mathbf{x}; \boldsymbol{\theta})$
    $\mathcal{L} \leftarrow l(\hat{\mathbf{y}}, \mathbf{y}) + \beta \cos^2(\mathrm{g}(0; \phi), \mathrm{g}(1; \phi))$    ▷ regularization to encourage diversity between endpoints
    Backprop from $\mathcal{L}$, update $\phi$ with $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}} = \frac{\partial l}{\partial \boldsymbol{\theta}} \frac{\partial \mathrm{g}}{\partial \boldsymbol{\phi}} + \beta \frac{\partial \cos^2(\mathrm{g}(0;\phi), \mathrm{g}(1;\phi))}{\partial \phi}$.
end for
```

Through the formulation in Algorithm 2, we consider $\mathrm{g}(0; \phi)$ and $\mathrm{g}(1; \phi)$ as the endpoints of an arbitrary 1-dimensional subspace in weight space $\mathbb{R}^{\boldsymbol{\omega}}$ parameterized by $\alpha \in [0, 1]$. The nonlinear mapping $\mathrm{g} : \alpha \rightarrow \mathbb{R}^{\boldsymbol{\omega}}$ is achieved with an MLP, and optimized via Algorithm 2. Our training objective then becomes an augmented version of Equation 1

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[\mathbb{E}_{\alpha\sim\text{Uniform}(0,1)}[l(f(\mathbf{x}, \mathrm{g}(\alpha; \phi), \mathbf{y})]] \tag{2}$$

where $\mathcal{D}$ denotes the data distribution, f denotes the function mapping between $\mathbf{x}$ and $\mathbf{y}$, and l denotes a loss function. Again, we optimize Equation 2 through stochastic updates and backpropagation, with additional cosine similarity regularization of the endpoints to ensure the subspace spans a reasonable region of weight space. This optimization procedure results in a parameterized 1-dimensional subspace $\mathrm{g}(\alpha; \phi)$ for which low loss is achieved throughout. Analysis of these learned non-linear subspaces could yield some insight about the structure and geometry of neural network loss spaces; we explore these concepts in this paper.

3.2 Subspace Analysis

Visualization:

Our goal through visualization was to distill some geometric understanding of our learned subspaces. We employ a range of techniques to visualize the learned loss subspaces. Each technique centered around finding a hyperplane through loss space, sampling from it and plotting the resultant loss or accuracy.

The simplest subspaces to visualize are the 2-simplex subspaces. These spaces are defined by three learned points $\{\boldsymbol{\omega}_i\}_{i=1}^3$ in weight space. It follows that the plane

$$S = \{\lambda_1 \boldsymbol{\omega}_1 + \lambda_2 \boldsymbol{\omega}_2 + \lambda_3 \boldsymbol{\omega}_3 \mid \lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}\}$$

spans the entire subspace, making it a natural choice for a set to sample from.

1-simplex subspaces unfortunately don't offer a natural solution like 2-simplexes. 1-simplexes are defined by their two endpoints $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2$, but a third point is required to constrain the set of intersecting hyperplanes to 1 element. To this end, we pull from Garipov et al. [2018], who obtain a third point in weight space via independently training a separate network for the same task. These three points span a single hyperplane through weight space that spans our 1-simplex and obtains low loss in another region.

Determining an appropriate hyperplane for visualizing the non-linear subspace required a similar approach. Our approach included sampling from the plane spanned by the two endpoints $\mathtt{g}(0;\phi), \mathtt{g}(1;\phi)$ and an independently trained network, a method analogous to the one described for 1-simplexes. We also played around with sampling from planes constrained by a combination of points in the subspace and principal components of points sampled from the subspace.

Perturbation Analysis:

In order to test the sensitivity of the learned subspaces as well as the visualizations, we perturb the endpoints of the learned subspaces for the 1-simplex case with varying amounts of Gaussian noise.

Dimensionality Estimation:

We also seek to estimate the inherent dimensionality (ID) of these learned subspaces, with the ambient space being the high dimensional weight space of our models. In this work, we use two methods to estimate ID:

1. Estimate ID through approximating the tangent plane with the rank of the matrix formed from the distance vectors of the k-nearest neighbors to a sampled point.
2. Estimate ID with a maximum likelihood estimator based on a Poisson Process approximation [Levina and Bickel, 2004]. This is done by growing spheres of various radii around a sampled point and observing how the volume of said sphere changes.

For both methods we take the average estimate across all sampled points of a chosen subspace.

Curvature Estimation:

A primary goal of ours is to analyze the curvature of our nonlinear subspace $\mathtt{g} : \alpha \to \mathbb{R}^{\boldsymbol{\omega}}$. We calculated values of curvature for sampled points along our subspace by performing a discrete approximation to the following arc-length definition for curvature,

$$\kappa(\alpha) = \left\| \frac{\partial \vec{T}(\alpha)}{\partial s(\alpha)} \right\| \tag{3}$$

where $\vec{T}(\alpha)$ is the tangent vector to our subspace about $\mathtt{g}(\alpha;\phi)$ and s is arc length.

Performance Comparison:

To evaluate the benefit of nonlinearity in subspace parameterizations we sought to compare the performance linear and nonlinear subspaces spanning the same regions of weight space. To do this we randomly sampled from our learned nonlinear subspace model $\mathtt{g}(\alpha;\phi)$ and compared the performance to sampling from a linear interpolation between the two endpoints $\mathtt{g}(0;\phi), \mathtt{g}(1;\phi)$ of the same subspace.

4 Experiments and results

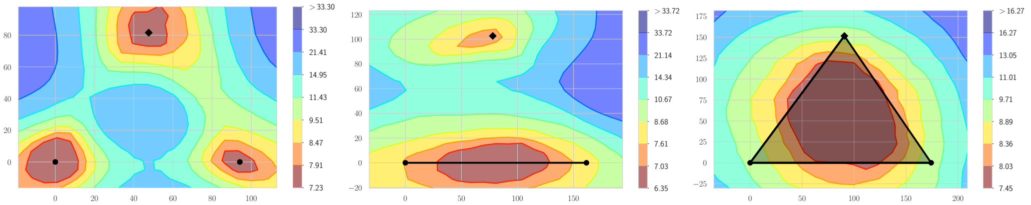
4.1 Experimental setup

In all of our experiments we trained a one hidden layer MLP, denoted by $f(\mathbf{x};\theta)$, on the Fashion MNIST dataset for image classification [Xiao et al., 2017], using 50,000 examples for training and 10,000 for validation. All models were trained for 50 epochs with batch sizes of 128 examples using the AdamW optimizer [Loshchilov and Hutter, 2019]. Dropout [Srivastava et al., 2014] after the hidden layer was performed and the rate was set to 0.3. This was all implemented using the `PyTorch` library [Paszke et al., 2019].

After training and validating that there is indeed low loss across the learned subspace, we perform analysis using the methods described in 3.2. Our model and dataset choices result in a weight space of 407,050 dimensions.

4.2 Affine Subspaces

Figure 1: 0-, 1-, and 2- simplexes visualized using plane method described in 3.2. Color gradient represents test error rate.



Results of training, evaluation and visualization of the 1- and 2- simplexes are shown in Figure 1, and demonstrate the effectiveness of the approach - it successfully finds a line in weight space that lies within a flat and low-loss portion of the optimization landscape. The visualization for the 2-simplex also demonstrates this, albeit with endpoints that occupy higher test error regions of the landscape compared to the 1-simplex. Another interesting obsevation from these plots are that even though all structures plotted in these spaces are in the form of simplexes, the resulting topography of loss are round and relatively flat.

Table 1: Dimensionality estimations for 1- and 2- simplex subspaces

Subspace Type	k-NN ID estimate	MLE ID estimate
1-simplex	1.00	1.17
2-simplex	1.82	1.84

Table 1 shows dimensionality estimates for the 1- and 2- simplex subspaces using the two methods described in 3.2. These estimates line up with you'd expect as the algorithms were able to identify the 1-simplex in the high dimensional ambient weight space as roughly 1 dimensional, while similarly they were also able to identify the 2-simplex as roughly 2 dimensional. This helps validate that our algorithm finds subspaces in the form of the shapes we expect them to be.

Figure 2: Perturbation analysis of line subspace. From left to right the plots show 1-simplex endpoints with no added noise, $\mathcal{N}(0, .01)$ added noise, and $\mathcal{N}(0, 1)$ added noise.

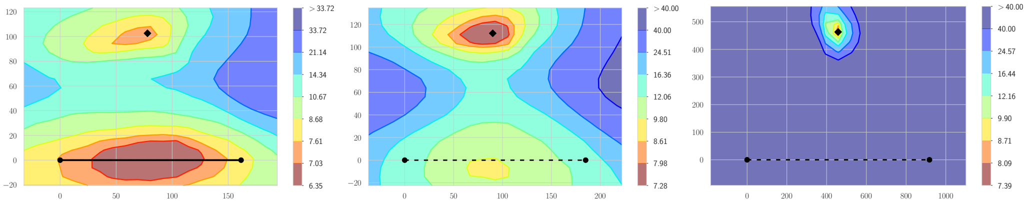


Figure 2 shows the results on visualization and test error rate of perturbing the 1-simplex endpoints with increasing amounts of Gaussian noise. Overall we see that these endpoints are quite sensitive to perturbations, demonstrating the minuteness of scale of these subspaces as a perturbation can take you from a region of low loss to high loss. This result also illustrates the importance of having a reasonably selected learning rate when optimizing your neural network.

4.3 Nonlinear Subspaces

Our method for finding effective nonlinear parameterizations consistently converged to low-loss regions of weight space. Sampling from these learned subspaces resulted in parameter settings θ

for which our classification network f achieved the same performance as an independently trained network. Visualizations in Figure 4.3 confirm the effectiveness of the optimization procedure, with the subspace network and independently trained neural network often converging to the same loss basin in weight space.

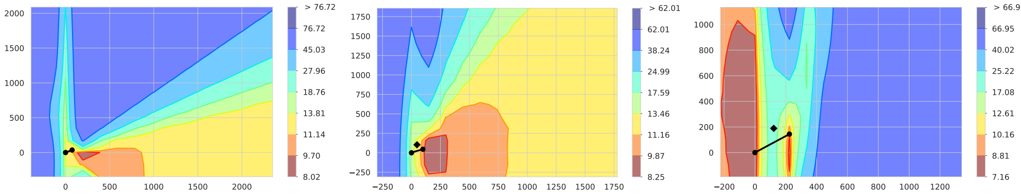


Figure 3: Test accuracy on parameter settings sampled from a hyperplane spanning our learned nonlinear subspace across three different seeds. Line and circular markers represent the learned subspace, diamond markers represent an independently trained neural network. The hyperplane was obtained via the approach detailed in Section 3.2.

We also sought to understand the geometry of these learned subspaces, as nonlinearity in the subspace parameterization introduces arbitrary complexity. Cursory experiments to compute dimensionality estimations confirmed that our learned subspace $\{\mathrm{g}(\alpha;\phi) \mid \alpha \in [0,1]\}$ had intrinsic dimensionality 1.

Of principal concern to us was the curvature of these parameterizations. While the subspace visualizations in 4.3 may appear to be linear, recall that these are merely projections of the subspace onto a hyperplane and likely mask much of intrinsic structure of the space. To understand these spaces to a higher degree, we compute discrete approximations of curvature along our subspace as mentioned in 3.2.

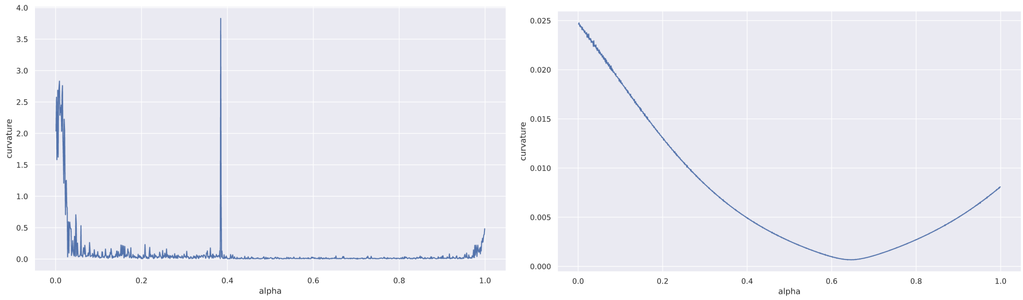


Figure 4: Estimated curvature at different points along our subspace, indicated by α. The subspace on the left is parameterized by a shallow MLP with ReLU nonlinearity, while the subspace on the right is parameterized by a deeper MLP with tanh nonlinearity.

Interestingly, we consistently find that our learned subspaces exhibit higher degrees of curvature near the edges of the space ($\alpha = 0, \alpha = 1$), a finding consistent with earlier work that analyzes the implicit biases of Stochastic Gradient Descent methods Damian et al. [2023]. This seems to suggest the optimization pushes the subspace to basins in the loss landscape, where the edges of the learned space $\mathrm{g}(0;\phi), \mathrm{g}(1;\phi)$ lie on the edges of the basins (which are of course higher curvature areas of the space). It is worth noting that spiking curvature estimates appear in situations where we use ReLU nonlinearities as opposed to tanh; discontinuous operators embedded in the neural network unsurprisingly result in discontinuities in our learned subspaces.

We also extended our analysis to experimenting with linear projections in an effort to gain more interpretable information about the subspace. Figure 4.3 shows the results of linearly projecting our nonlinear subspace onto estimates of its principal components across three different seeds. These estimates were computed by a mixture of sampling and Principal Component Analysis (PCA). The plots verify our findings from Figure 3 concerning the non-zero curvature of our subspaces.

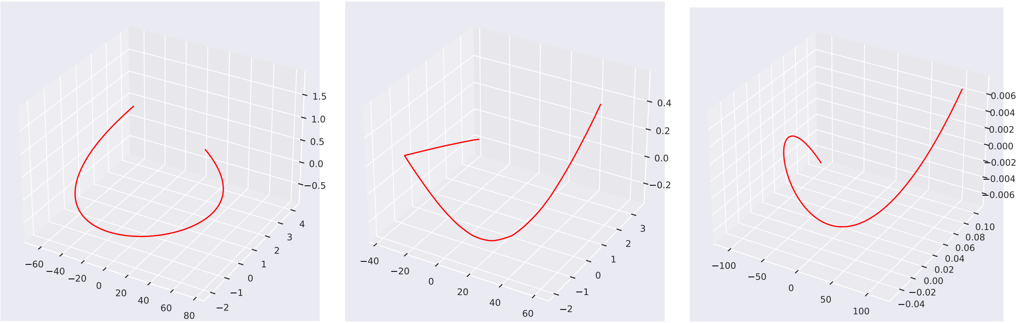


Figure 5: Projection of our nonlinear subspace onto its three principal directions for various seeds. Estimates of these principal components were obtained via sampling $\mathrm{g}(\alpha;\phi), \alpha \sim \mathrm{Uniform}[0,1]$ and performing Principal Component Analysis (PCA) on the sampled points.

While this analysis implies that our method has successfully found nonlinear regions of low-loss in weight space, the effect and necessity of introducing nonlinearity is not yet clear. It is certainly possible that our optimization procedure yielded subspaces nestled in vast basins of low loss. In this case, simple interpolation between the endpoints of the space would be just as effective. Unfortunately in testing this hypothesis, we found that samples from linear interpolation between endpoints $\mathrm{g}(0;\phi), \mathrm{g}(1;\phi)$ of our learned subspace to perform equally well as samples directly from our subspace itself in a majority of cases. While at a glance this may suggest that simpler models are sufficient for subspace learning purposes, we believe that future extensions of this project could disprove that idea.

5 Conclusion

These results suggest that more extensive analysis could open the door to a deeper understanding of loss landscapes. Evidently, fitting linear subspaces to the loss landscape produces effective ensembling methods and interesting visualizations. However, it does little by way of expanding our understanding the geometry of these spaces. Our push to fit nonlinear, 1-dimensional subspaces to the loss landscape has been a fruitful push, but more than anything it has yielded information that will inform our future efforts. We now believe that higher dimensional subspaces and intentional selection of the subspace-fitting loss function to encourage span could result in subspace fitting that captures more information about the nature of the loss landscape.

References

Gregory W Benton, Wesley J Maddox, Sanae Lotfi, and Andrew Gordon Wilson. Loss surface simplexes for mode connecting volumes and fast ensembling. In *International Conference on Machine Learning*, 2021.

Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability, 2023.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. 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 1309–1318. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/draxler18a.html`.

Stanislav Fort and Stanislaw Jastrzebski. *Large Scale Structure of Neural Network Loss Landscapes*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective, 2019. URL `https://arxiv.org/abs/1912.02757`.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8803–8812, Red Hook, NY, USA, 2018. Curran Associates Inc.

Justin Gilmer, Behrooz Ghorbani, Ankush Garg, Sneha Kudugunta, Behnam Neyshabur, David Cardoze, George Edward Dahl, Zachary Nado, and Orhan Firat. A loss curvature perspective on training instabilities of deep learning models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=OcKMT-36vUs`.

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004. URL `https://proceedings.neurips.cc/paper_files/paper/2004/file/74934548253bcab8490ebd74afed7031-Paper.pdf`.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019. URL `http://arxiv.org/abs/1912.01703`.

Ivan Skorokhodov and Mikhail Burtsev. Loss landscape sightseeing with multi-point optimization, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

Mitchell Wortsman, Maxwell C Horton, Carlos Guestrin, Ali Farhadi, and Mohammad Rastegari. Learning neural network subspaces. 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 11217–11227. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/wortsman21a.html`.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.