Using Mode Connectivity for Loss Landscape Analysis

Akhilesh Gotmare [1] **Nitish Shirish Keskar** [1] **Caiming Xiong** [1] **Richard Socher** [1]

Abstract

Mode connectivity is a recently introduced framework that empirically establishes the connectedness of minima by finding a high accuracy curve between two independently trained models. To investigate the limits of this setup, we examine the efficacy of this technique in extreme cases where the input models are trained or initialized differently. We find that the procedure is resilient to such changes. Given this finding, we propose using the framework for analyzing loss surfaces and training trajectories more generally, and in this direction, study SGD with cosine annealing and restarts (SGDR). We report that while SGDR moves over barriers in its trajectory, propositions claiming that it converges to and escapes from multiple local minima are not substantiated by our empirical results.

1. Introduction and Related Work

Training neural networks involves optimizing a non-convex objective function with gradient-based methods. Recent work focused on understanding the loss surface of neural networks and the trajectories traced by optimizers like stochastic gradient descent (SGD) and its adaptive variants, including Adam (Kingma & Ba, 2014), Adagrad (Duchi et al., 2011), and RMSProp (Tieleman & Hinton, 2012).

Garipov et al. (2018) introduce a framework to obtain a low loss (or high accuracy, in the case of classification) curve of simple form, such as a piecewise linear curve, that connects optima (modes of the loss function) found independently. This observation suggests that, unlike several empirical results claiming that minima are isolated or have barriers between them, these points, in fact, are connected, at the same loss function depth, via a simple piecewise linear curve. Draxler et al. (2018) independently report the same observation for neural network loss landscapes, and claim that this is suggestive of the resilience of neural networks to perturbations in model parameters.

In this work, we present two novel results: first, we evaluate the resilience of the mode connectivity phenomenon by using the proposed procedure to connect optima found via different training schemes, and then proceed to use it as a tool to make observations on the optimization trajectory of SGD with cosine-annealing (SGDR) (Loshchilov & Hutter, 2016). We study this heuristic in particular given its superior empirical performance on many tasks, see e.g., (Coleman et al., 2017), and also the lack of theoretical motivation explaining why it works.

We begin by briefly describing the mode connectivity procedure in Section 2, and the SGDR strategy in Section 4. In Section 3, we present a short motivation, experimental details and results for testing the mode connectivity framework's resilience. Section 5 involves the experiments and analysis of the loss surface and SGDR trajectory.

2. Mode Connectivity Procedure

Let $w_a \in \mathbb{R}^D$ and $w_b \in \mathbb{R}^D$ be two modes (optimal sets of neural network parameters) in the D-dimensional parameter space obtained using independent training runs that both optimize a given loss function $\mathcal{L}(w)$ (like the cross-entropy loss). We represent a curve connecting w_a and w_b by $\phi_\theta(t) : [0,1] \rightarrow \mathbb{R}^D$, such that $\phi_\theta(0) = w_a$ and $\phi_\theta(1) = w_b$. To find a low loss path, we find the set of parameters $\theta \in \mathbb{R}^D$ that minimizes the following loss:

$$\ell(\theta) = \int_0^1 \mathcal{L}(\phi_\theta(t))dt = \mathbb{E}_{t\sim U(0,1)}\mathcal{L}(\phi_\theta(t))$$

where $U(0,1)$ is the uniform distribution in the interval $[0,1]$.

To optimize $\ell(\theta)$ for θ, we first need to chose a parametric form for $\phi_\theta(t)$. One of the forms proposed by Garipov et al. (2018) is a polygonal chain with a single bend at θ as follows

$$\phi_\theta(t) = \begin{cases} 2(t\theta + (0.5-t)w_a), & \text{if } 0 \leq t \leq 0.5 \\ 2((t-0.5)w_b + (1-t)\theta) & \text{if } 0.5 < t \leq 1 \end{cases}$$

[1] Salesforce Research, Palo Alto, USA. Correspondence to: Nitish Shirish Keskar <nkeskar@salesfoce.com>.

Part of 35th International Conference on Machine Learning's Workshop on Modern Trends in Nonconvex Optimization for Machine Learning, Stockholm, Sweden, 2018. Copyright 2018 by the author(s).

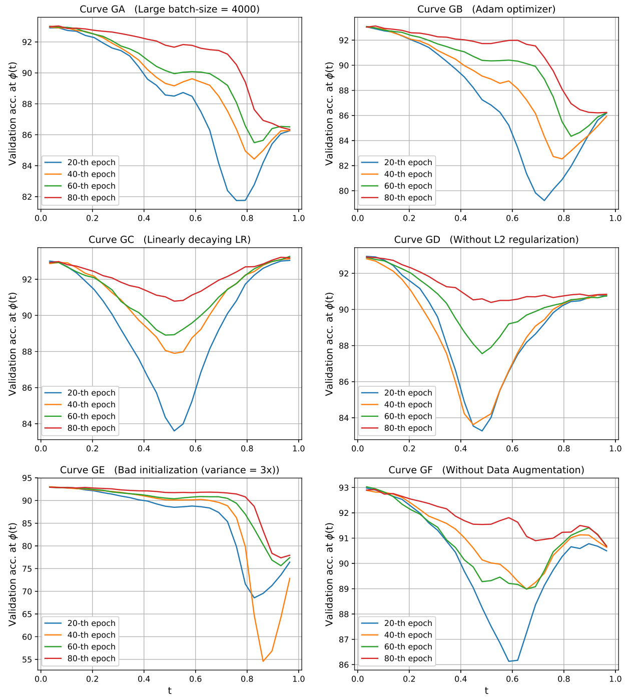


Figure 1. Validation accuracy corresponding to models on the following 6 different curves - curve GA represents curve connecting mode G (one found with all the default hyperparameters) and mode A (found using large batch size), similarly, curve GB connects mode G and mode B (found using Adam), curve GC connects to mode C (found using linearly decaying LR), curve GD to mode D (found with much less L2 regularization), curve GE to mode E (found using a poor initialization), and curve GF to mode F (found without using data augmentation). $t = 0$ corresponds to mode G for all of the plots.

To minimize $\ell(\theta)$, we sample $t \sim U[0,1]$ at each iteration and use the quantity $\nabla_\theta \mathcal{L}(\phi_\theta(t))$ as an estimate for the true gradient $\nabla_\theta \ell(\theta)$ to perform updates on θ (using SGD). We initialize θ with $\frac{1}{2}(w_a + w_b)$. Note that in expectation over the uniformly distributed t, this computationally cheap estimate is equal to the true gradient

$$\mathbb{E}_{t\sim U[0,1]} \nabla_\theta \mathcal{L}(\phi_\theta(t)) = \nabla_\theta \mathbb{E}_{t\sim U[0,1]} \mathcal{L}(\phi_\theta(t)) = \nabla_\theta(\ell(\theta))$$

3. Resilience of Mode Connectivity

To demonstrate that the curve-finding approach described in Section 2 works in practice, Garipov et al. (2018) use two optima found using different initializations but a common training scheme which we detail below. We explore the limits of this procedure by connecting optima obtained from different training strategies. In particular, we experiment with different initializations, optimizers, data augmentation choices, and hyperparameter settings including regularization, training batch sizes and learning rate schemes. Conventional wisdom suggests that these different training schemes will converge to different regions in the parameter space that are isolated from each other. Having a high accuracy connection between these pairs would seem counterintuitive. Particularly for the large batch training case, previous works (Hochreiter & Schmidhuber, 1997; Keskar et al., 2016) have empirically established that small-batch training leads to wider minima and large-batch training leads to sharper minima. Likewise, with respect to models found using different optimizers, Heusel et al. (2017) argues that Adam also prefers wide optima and Wilson et al. (2017) show that adaptive methods like Adam lead to drastically different solutions from SGD. Similarly, in the context of importance of initialization, Goodfellow et al. (2016) show that the scale of the distribution used for initialization has a large impact on both the outcome of the optimization algorithm and the ability of the network to generalize. Lastly, we know that L-2 regularization or weight decay drives the parameters closer to 0, while a smaller L-2 penalty would have a lesser effect of this kind and thus would allow the optimization path to explore models farther from the origin.

For obtaining the reference model (named mode G), we train the VGG16 model architecture (Simonyan & Zisserman, 2014) using CIFAR10 training data (Krizhevsky et al., 2014) for 200 epochs with SGD. The learning rate is initialized to 0.05 and scaled down by a factor of 5 at epochs $\{60, 120, 160\}$ (step decay). We use a training batch size of 100, momentum of 0.9, and a weight decay of 0.0005. Elements of the weight vector corresponding to a neuron are initialized randomly from the normal distribution $\mathcal{N}(0, \sqrt{2/n})$ where n is the number of inputs to the neuron. We also use data augmentation by random cropping of input images.

We build 6 variants of the reference mode G as follows: we obtain mode A using a training batch size of 4000, mode B by using the Adam optimizer instead of SGD (and hence also a different learning rate), mode C with a linearly decaying LR scheme instead of the step decay scheme used in mode G, mode D using a smaller weight decay of 5×10^{-6}, mode E by increasing the variance of the initialization distribution to $3\times\sqrt{2/n}$ and mode F using no data augmentation.

Note that for this set of modes $\{A, B, C, D, E, F\}$ all the other hyper-parameters and training settings except the ones mentioned above are the same as that for mode G. We use the mode connectivity algorithm on each of the 6 pairs of modes including G and another mode, resulting in curves GA, GB, GC, GD, GE, and GF.

Figure 1 shows the validation accuracy for models on each of the 6 connecting curves during the 20th, 40th, 60th and 80th epochs of the mode connectivity training proce-

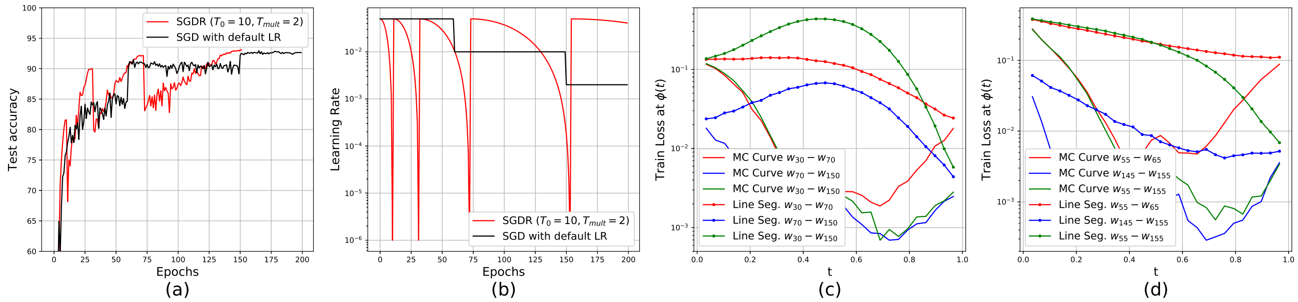


Figure 2. (a) Validation accuracy of a VGG16 model trained on CIFAR10 using SGDR with warm restarts simulated every $T_0 = 10$ epochs and doubling the periods T_i at every new warm restart ($T_{mult} = 2$). (b) SGDR and SGD learning rate schemes. (c) Cross-entropy training loss on the curve found through Mode Connectivity (MC Curve) and on the line segment (Line Seg.) joining modes w_{30} (model corresponding to parameters at the 30-th epoch of **SGDR**) and w_{70}, w_{70} and w_{150}, w_{30} and w_{150}. (d) Cross-entropy training loss on the curve found through Mode Connectivity (MC Curve) and on the line segment (Line Seg.) joining modes w_{55} (model corresponding to parameters at the 55-th epoch of **SGD with step decay LR scheme**) and w_{65}, w_{145} and w_{155}, w_{55} and w_{155}.

dure. As described in Section 2, for a polychain curve GX (connecting modes G and X using the curve described by θ), model parameters $\phi_\theta(t)$ on the curve are given by $p_{\phi_\theta(t)} = 2(tp_\theta + (0.5 - t)p_G)$ if $0 \leq t \leq 0.5$ and $p_{\phi_\theta(t)} = 2((t - 0.5)p_X + (1 - t)p_\theta)$ if $0.5 < t \leq 1$ where p_G, p_θ and p_X are parameters of the models G, θ, and X respectively. Thus $\phi_\theta(0) = G$ and $\phi_\theta(1) = X$.

Within a few epochs of the curve training, for each of the 6 pairs, we can find a curve such that each point on it generalizes almost as well as models from the pair that is being connected. Thus we are able to find a high-accuracy connection between each of the 6 pairs. While connectivity for the pairs $\{G, C\}$ and $\{G, F\}$ might not be particularly surprising, one would expect the other cases to be isolated from each other and divided by high loss regions. Note that by virtue of existence of these 6 curves, there exists a high accuracy connecting curve (albeit with multiple bends) for each of the ${}^7C_2$ pairs of modes. We refer the reader to Figure 4 in the Appendix for a t-SNE plot of the modes and their connections.

Having established the high likelihood of the existence of these connecting curves, we use the curve finding procedure along with interpolating loss surface between parameters at different epochs as tools to analyze the dynamics of SGD and SGD with warm restarts (SGDR).

4. SGD with Warm Restarts

Loshchilov & Hutter (2016) introduced SGDR as an interesting modification to the cyclical LR scheme (Smith, 2017) that combines restarts with cosine annealing. The learning rate at the t-th epoch in SGDR is given by the following expression in (1) where η_{min} and η_{max} are the lower and upper bounds respectively for the LR. T_{cur} represents how many epochs have been performed since the last restart and a warm restart is simulated once T_i epochs are performed. Also $T_i = T_{mult} \times T_{i-1}$, meaning the period T_i for the LR variation is increased by a factor of T_{mult} after each restart. Figure 2 (b) shows an instance of this LR schedule.

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{T_{cur}}{T_i}\pi\right)\right) \quad (1)$$

The model returned at the end of training is the one corresponding to the iterate at the epoch just before the last restart (epoch 150 in Figure 2 (b)).

While the strategy has been claimed to outperform other LR schedulers, little is known why this has been the case. One explanation that has been given in support of SGDR is that it can be useful to deal with multi-modal functions, where our iterates could get stuck in a local optimum and a restart will help them get out of it and explore another region; however, Loshchilov & Hutter (2016) do not claim to observe any effect related to multi-modality. Huang et al. (2017) propose an ensembling strategy using the set of iterates before restarts and claim that, when using the learning rate annealing cycles, the optimization path converges to and escapes from several local minima. In the next section, we try to empirically investigate if this is actually the case by interpolating the loss surface between parameters at different epochs and studying the training and validation loss for parameters on the plane passing through the two modes found by SGDR and their connectivity (plane defined by affine combinations of w_a, w_b and θ).

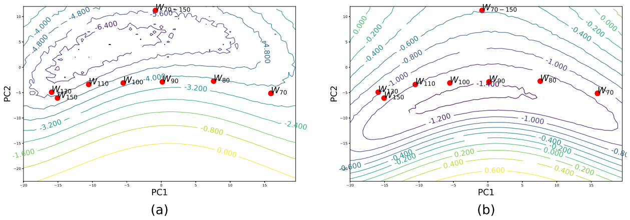
5. Loss Surface Analysis with Mode Connectivity

We train a VGG16 network on the CIFAR10 dataset for image classification using SGD with warm restarts (SGDR). For our experiments, we choose $T_0 = 10$ epochs and $T_{mult} = 2$ (warm restarts simulated every 10 epochs and the period T_i doubled at every new warm restart), $\eta_{max} =$

0.05 and $\eta_{min} = 10^{-6}$. We also perform VGG training using SGD (with momentum of 0.9) and a step decay LR scheme (initial LR of $\eta_0 = 0.05$, scaled by 5 at epochs 60 and 150). Figure 2 (b) shows the LR variation for these two schemes on a logarithmic scale and Figure 2 (a) shows the validation accuracy over training epochs with these two LR schemes.

In order to understand the loss landscape on the optimization path of SGDR, the pairs of iterates obtained just before the restarts $\{w_{30}, w_{70}\}, \{w_{70}, w_{150}\}$ and $\{w_{30}, w_{150}\}$ are given as inputs to the mode connectivity algorithm, where w_n is the model corresponding to parameters at the n-th epoch of the SGDR training. Figure 2 (c) shows the training loss for models along the line segment joining these pairs and those on the curve found through mode connectivity. For the baseline case, we connect the iterates around the epochs when we decrease our LR in the step decay LR scheme. Thus we chose $\{w_{55}, w_{65}\}, \{w_{145}, w_{165}\}$ and $\{w_{55}, w_{165}\}$ as input pairs to the mode connectivity algorithm where now w_n is the model corresponding to parameters at the n-th epoch of SGD with the step decay LR scheme. Figure 2 (d) shows the training loss for models along the line segments joining these pairs and the curves found through mode connectivity.

From Figure 2 (c), it is clear that for the pairs $\{w_{30}, w_{150}\}$ and $\{w_{70}, w_{150}\}$ the training loss for points on the line segment is much higher than the endpoints suggesting that SGDR indeed finds paths that move over a barrier[1] in the training loss landscape. In contrast, for SGD (without restarts) in Figure 2 (d) none of the three pairs show evidence of having a training loss barrier on the line segment joining them. Instead there seems to be an almost linear decrease of training loss along the direction of these line segments, suggesting that SGD's trajectory is quite different from SGDR's. We present additional experiments, including results for other metrics, in Appendix A.4.



(a)

(b)

Figure 3. (a) Training loss surface (log scale) and (b) validation loss surface (log scale) for points (models) on the plane defined by $\{W_{70}, W_{150}, W_{70-150}\}$ including projections of the SGDR iterates on this plane

To understand the SGDR trajectory more concretely, we evaluate the intermediate iterates on the plane in the D-dimensional space defined by the three points: w_{70}, w_{150} and w_{70-150}, where w_{70-150} is the θ that defines the high accuracy connection for the pair $\{w_{70}, w_{150}\}$. This 2-d plane in the D-dimensional space consists of all the affine combinations of w_{70}, w_{150} and w_{70-150}. Figure 3 (a) and 3 (b) show the training and validation loss surface for points in this subspace, respectively. Note that the intermediate iterates do not necessarily lie in this plane and thus need to be projected. Hence, one cannot tell the value of loss at the actual iterates from their representation in Figure 3. We refer the reader to Appendix A.2 for additional details regarding the projection process and Appendix A.3 for analogous results with w_{30} and w_{70}.

Figure 3 (a) suggests that SGDR helps the iterates to converge to a different region although neither of w_{70} or w_{150} are technically a local minimum, nor do they appear to be lying in different *basins*, hinting that Huang et al. (2017)'s claims about SGDR converging to and escaping from local minima might be an oversimplification.

Another insight we can draw from Figure 3 (a) is that the path found by mode connectivity corresponds to lower training loss than the loss at the iterates that SGDR converges to ($\mathcal{L}(w_{150}) > \mathcal{L}(w_{70-150})$). However, Figure 3 (b) shows that models on this curve seem to overfit and not generalize as well as the iterates w_{70} and w_{150} which stands as further evidence that SGD's stochasticity helps generalization. This observation is also consistent with what we see in Figure 1. Thus, gathering models from this connecting curve might seem as a novel and computationally cheap way of creating ensembles, this generalization gap alludes to one limitation in doing so; Garipov et al. (2018) point to other shortcomings of curve ensembling in their original work.

In Figure 3, the region of the plane under consideration, between the iterates w_{70} and w_{150}, corresponds to higher training loss but lower validation loss than the two iterates. This hints at a reason why averaging iterates to improve generalization using cyclic or constant learning rates (Izmailov et al., 2018) has been found to work reasonably well.

6. Conclusion

We revisited the recently proposed mode connectivity procedure, and explored its limits by using it to find the desired connection between models trained with different training schemes and initializations. Remarkably, we found curves with reasonably high accuracy for mode pairs that we considered. These results are indicative of the connectedness of deep learning loss surface minima. Given this resiliency,

[1] a path is said to have moved over or crossed a barrier between epoch m and n ($n > m$) if $\exists\, w_t \in \{\lambda w_m + (1-\lambda) w_n | \lambda \in [0,1]\}$ such that $\mathcal{L}(w_t) > \max\{\mathcal{L}(w_m), \mathcal{L}(w_n)\}$

we use the framework to inspect the claims made to explain the effectiveness of restarts and cosine annealing in SGDR, and studied the SGDR trajectory using the subspace defined by the mode connections. We found that although SGDR tends to move over barriers, claims about SGDR converging to and escaping multiple local minima are not substantied by our experiments. Our work establishes the wide generality of the mode connectivity framework, and encourages use of it as a tool for understanding not just the training landscape but also the training trajectories.

References

Coleman, Cody, Narayanan, Deepak, Kang, Daniel, Zhao, Tian, Zhang, Jian, Nardi, Luigi, Bailis, Peter, Olukotun, Kunle, Ré, Chris, and Zaharia, Matei. Dawnbench: An end-to-end deep learning benchmark and competition. *Training*, 100(101):102, 2017.

Draxler, Felix, Veschgini, Kambis, Salmhofer, Manfred, and Hamprecht, Fred A. Essentially no barriers in neural network energy landscape. *arXiv preprint arXiv:1803.00885*, 2018.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Garipov, Timur, Izmailov, Pavel, Podoprikhin, Dmitrii, Vetrov, Dmitry P, and Wilson, Andrew Gordon. Loss surfaces, mode connectivity, and fast ensembling of dnns. *arXiv preprint arXiv:1802.10026*, 2018.

Goodfellow, Ian, Bengio, Yoshua, Courville, Aaron, and Bengio, Yoshua. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, Klambauer, Günter, and Hochreiter, Sepp. Gans trained by a two time-scale update rule converge to a nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

Hochreiter, Sepp and Schmidhuber, Jürgen. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Huang, Gao, Li, Yixuan, Pleiss, Geoff, Liu, Zhuang, Hopcroft, John E, and Weinberger, Kilian Q. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

Izmailov, Pavel, Podoprikhin, Dmitrii, Garipov, Timur, Vetrov, Dmitry, and Wilson, Andrew Gordon. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Keskar, Nitish Shirish, Mudigere, Dheevatsa, Nocedal, Jorge, Smelyanskiy, Mikhail, and Tang, Ping Tak Peter. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Kingma, Diederik P and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, Alex, Nair, Vinod, and Hinton, Geoffrey. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

Langley, P. Crafting papers on machine learning. In Langley, Pat (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Loshchilov, Ilya and Hutter, Frank. Sgdr: stochastic gradient descent with restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Maaten, Laurens van der and Hinton, Geoffrey. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

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

Smith, Leslie N. Cyclical learning rates for training neural networks. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pp. 464–472. IEEE, 2017.

Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, and Fergus, Rob. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tieleman, Tijmen and Hinton, Geoffrey. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

Wilson, Ashia C, Roelofs, Rebecca, Stern, Mitchell, Srebro, Nati, and Recht, Benjamin. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 4151–4161, 2017.

A. Additional Results

A.1. t-SNE visualization for the 7 modes

We use t-SNE (Maaten & Hinton, 2008) to visualize these 7 modes and the θ points that define the connectivity for the 6 pairs presented in Section 3, in a 2-dimensional plot in Figure 4. Since t-SNE is known to map only local information correctly and not preserve global distances, we caution the reader about the limited interpretability of this visualization, it is presented simply to establish the notion of connected modes.

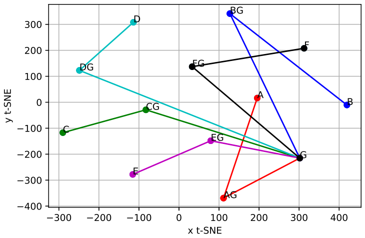


Figure 4. Representing the modes and their connecting point using t-SNE

It is interesting to note that although neural networks performance are quite resilient to changes in weight (and bias) parameters as suggested by the mode connectivity phenomenon, that is not the case when it comes to perturbations to the input of the network (Szegedy et al., 2013).

A.2. Projecting iterates

The W_n in Figure 3(a) and 3(b) is equivalent to

$$W_n = P_c(w_n) = \lambda^{\star\top} \begin{bmatrix} w_{70} \\ w_{150} \\ \theta \end{bmatrix}$$

$$\text{where } \lambda^\star = \text{argmin}_{\lambda\in\mathbb{R}^3} ||\lambda^\top \begin{bmatrix} w_{70} \\ w_{150} \\ \theta \end{bmatrix} - w_n||_2^2$$

meaning it is the point on the plane (linear combination of w_{70}, w_{150} and θ) with the least l-2 distance from the original point (iterate in this case).

A.3. Connecting modes W_{30} and W_{70} from SGDR

In Section 4, we present some experiments and make observations on the trajectory of SGDR by using the plane defined by the points w_{70},w_{150} and w_{70-150}. Here we provide the loss surface plots for another plane defined by SGDR's iterates $\{w_{30}, w_{70}, w_{30-70}\}$ to ensure the reader that the observations made are general enough.

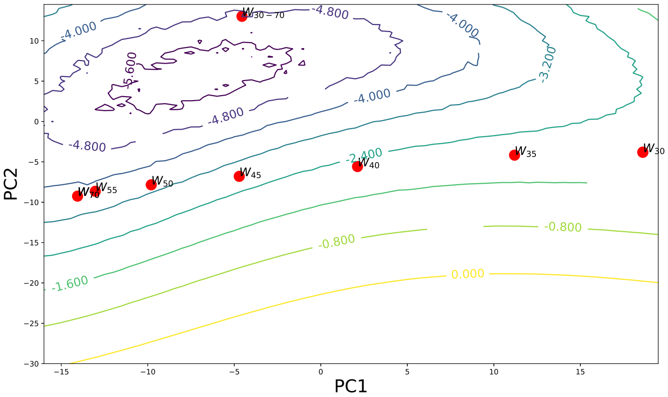


Figure 5. Training Loss Surface (log scale)

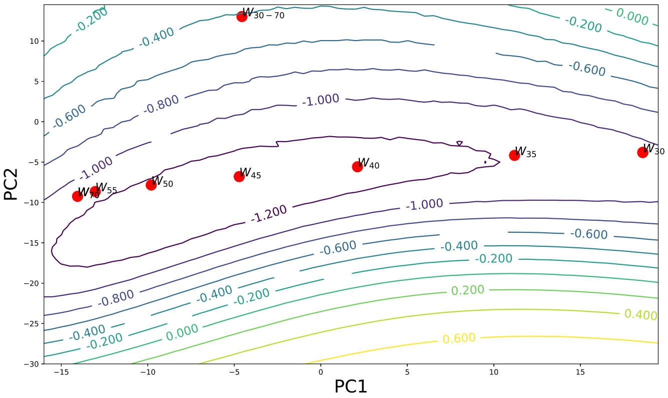


Figure 6. Validation Loss Surface (log scale)

A.4. Additional Experiments

For completeness, we present Figure 7 plotting the remaining quantities - namely Training Accuracy, Validation Accuracy and Validation Loss over the connecting curve from Section 5, Figure 2 for the pair of iterates obtained using SGD and SGDR.

Figures 8, 9 and 10 show the Validation Loss, Training Accuracy and Training Loss respectively for the curves joining the 6 pairs discussed in Section 3. These results too, confirm the overfitting or poor generalization tendency of models on the curve.

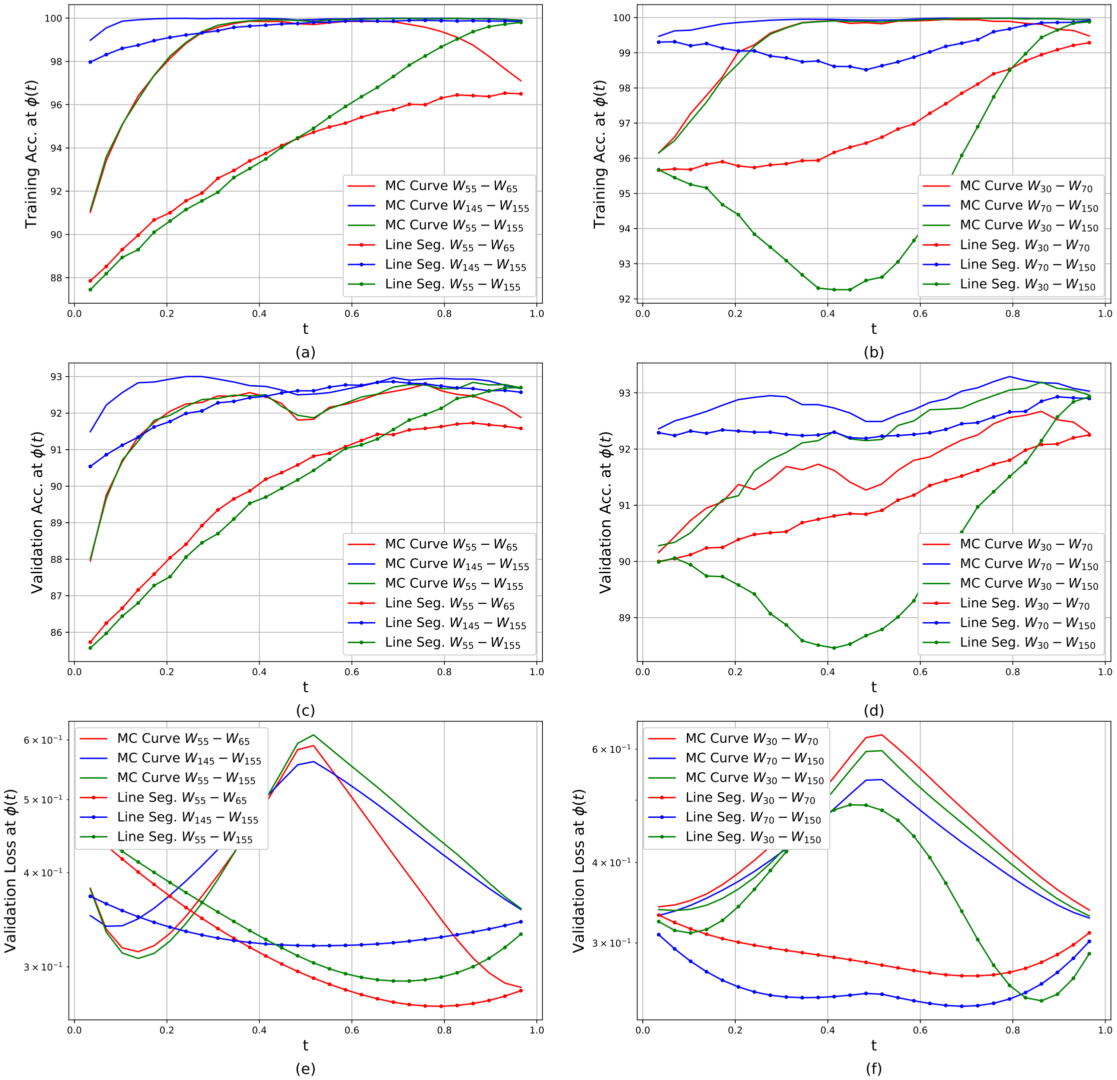


Figure 7. **Left** Column: Connecting iterates from SGD with step-decay LR scheme **Right** Column: Connecting iterates from SGDR **Top** Row: Training Accuracy on the curve found through Mode Connectivity (MC Curve) and on the line segment (Line Seg.) joining iterates from SGDR and SGD. **Middle** row: Validation Accuracy on the curve found through Mode Connectivity (MC Curve) and on the line segment (Line Seg.) joining iterates from SGDR and SGD. **Bottom** row Validation Loss on the curve found through Mode Connectivity (MC Curve) and on the line segment (Line Seg.) joining iterates from SGDR and SGD.

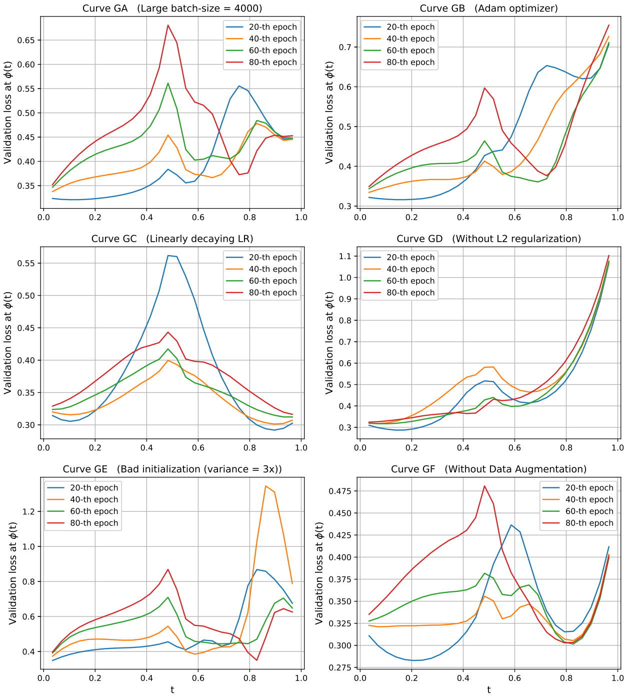


Figure 8. Validation Loss corresponding to models on the following 6 different curves - curve GA represents curve connecting mode G (one found with all the default hyperparameters) and mode A (found using large batch size), similarly, curve GB connects mode G and mode B (found using Adam), curve GC connects to mode C (found using linearly decaying LR), curve GD to mode D (found with much less L2 regularization), curve GE to mode E (found using a poor initialization), and curve GF to mode F (found without using data augmentation).

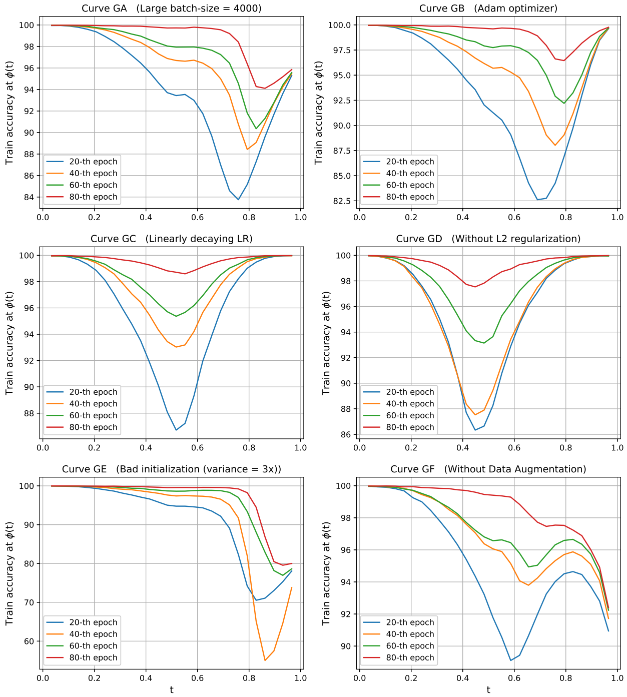


Figure 9. Training accuracy corresponding to models on the following 6 different curves - curve GA represents curve connecting mode G (one found with all the default hyperparameters) and mode A (found using large batch size), similarly, curve GB connects mode G and mode B (found using Adam), curve GC connects to mode C (found using linearly decaying LR), curve GD to mode D (found with much less L2 regularization), curve GE to mode E (found using a poor initialization), and curve GF to mode F (found without using data augmentation).

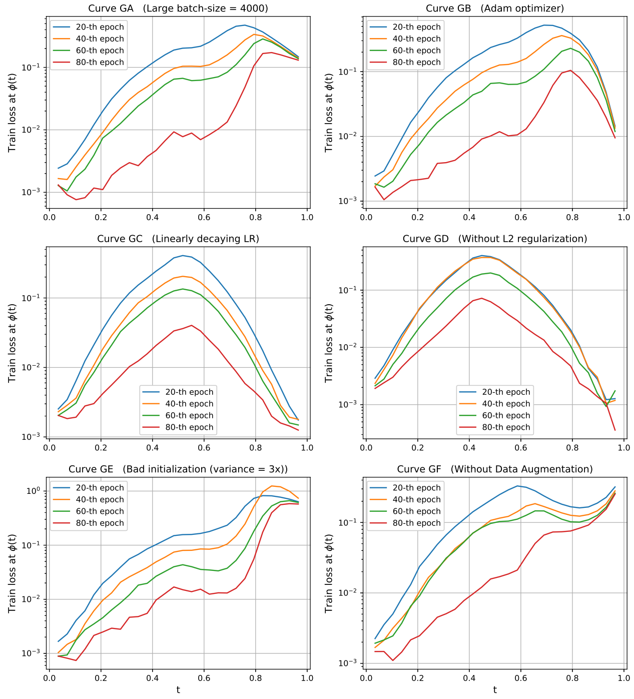


Figure 10. Training Loss corresponding to models on the following 6 different curves - curve GA represents curve connecting mode G (one found with all the default hyperparameters) and mode A (found using large batch size), similarly, curve GB connects mode G and mode B (found using Adam), curve GC connects to mode C (found using linearly decaying LR), curve GD to mode D (found with much less L2 regularization), curve GE to mode E (found using a poor initialization), and curve GF to mode F (found without using data augmentation).