
Asymmetric Valleys: Beyond Sharp and Flat Local Minima

Haowei He¹ Gao Huang² Yang Yuan³

Abstract

Despite the non-convex nature of their loss functions, deep neural networks are known to generalize well when optimized with stochastic gradient descent (SGD). Recent work conjectures that SGD with proper configuration is able to find wide and flat local minima, which have been proposed to be associated with good generalization performance. In this paper, we observe that local minima of modern deep networks are more than being flat or sharp. Specifically, at a local minimum there exist many asymmetric directions such that the loss increases abruptly along one side, and slowly along the opposite side – we formally define such minima as *asymmetric valleys*. Under mild assumptions, we prove that for asymmetric valleys, a solution biased towards the flat side generalizes better than the exact minimizer. Further, we show that simply averaging the weights along the SGD trajectory gives rise to such biased solutions implicitly. This provides a theoretical explanation for the intriguing phenomenon observed by Izmailov et al. (2018). In addition, we empirically find that batch normalization (BN) appears to be a major cause for asymmetric valleys.

1. Introduction

The loss landscape of neural networks has attracted great research interests in the deep learning community (Choromanska et al., 2015; Cooper, 2018; Keskar et al., 2017; Draxler et al., 2018; Ge et al., 2017; Sagun et al., 2017). It provides the basis of designing better optimization algorithms, and helps to answer the question of when and how a deep network can achieve good generalization performance. One hypothesis that draws attention recently is that the local minima of neural networks can be characterized by their flatness, and it is conjectured that sharp minima tend to generalize worse than the flat ones (Keskar et al., 2017). A plausible explanation is that a flat minimizer of the training

loss can achieve lower generalization error if the test loss is shifted from the training loss due to random perturbations. Figure 1(a) gives an illustration for this argument.

Although being supported by plenty of empirical observations (Keskar et al., 2017; Izmailov et al., 2018; Li et al., 2018), the definition of flatness was recently challenged by (Dinh et al., 2017), who showed that one can construct arbitrarily sharp minima through weight re-parameterization without changing the generalization performance. In addition, recent evidence suggests that the minima of modern deep networks are connected with simple paths with low generalization error (Draxler et al., 2018; Garipov et al., 2018). Similarly, the minima found by large batch training and small batch training are shown to be connected without any “bumps” (Sagun et al., 2017). This raises several questions: (1) If all the minima are well connected, why do some algorithms keep finding sharp minima and others keep finding flat ones (Keskar et al., 2017)? (2) Does flatness really affect generalization?

In this paper, we address these questions by introducing the concept of *asymmetric valleys*. We observe that the local geometry of the loss function of neural networks is usually asymmetric. In other words, there exist many directions such that the loss increases abruptly along one side, and grows rather slowly along the opposite side (see Figure 1(b) as an illustration). We formally define this kind of local minima as asymmetric valleys. As we will show in Section 6, asymmetric valleys brings interesting illusions in high dimensional space. For example, located in the same valley, \tilde{w} may appear to be a wider and flatter minimum than \hat{w} as the former is farther away from the sharp side.

For the second question, we argue that flatness does affect generalization. However, we do not simply follow the argument in (Keskar et al., 2017), which states that flat minima tend to generalize better because they are more stable. Instead, we prove that in asymmetric valleys, the solution biased towards the flat side of the valley gives better generalization under mild assumptions. This result has at least two interesting implications: (1) converging to *which* local minimum (if there are many) may not be critical for modern deep networks. However, it matters a lot *where* the solution locates; and (2) the solution with lowest *a priori* generalization error is not necessarily the minimizer of the training

¹Beihang University, China ²Tsinghua University, China
³MIT, United States. Correspondence to: Haowei He
<hehaowei@buaa.edu.cn>.

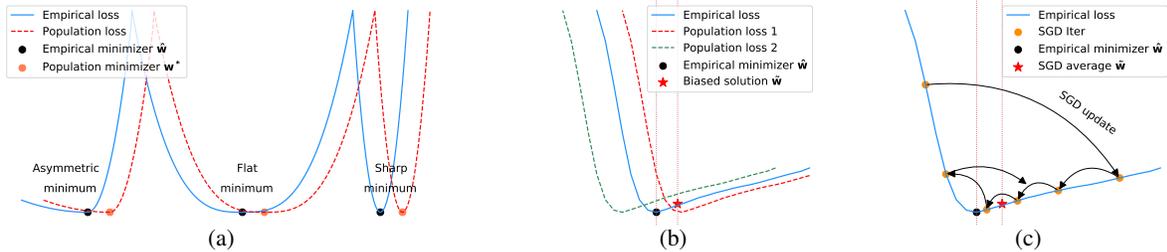


Figure 1. (a) Three kinds of local minima: asymmetric, flat and sharp. If there exists a shift from empirical loss to population loss, flat minimum is more robust than sharp minimum. (b) For asymmetric valleys, if there exists a random shift, the solution \tilde{w} biased towards the flat side is more robust than the minimizer \hat{w}^* . (c) SGD tends to stay longer on the flat side of asymmetric valleys, therefore SGD averaging automatically produces the desired bias.

loss.

Given that a biased solution is preferred for asymmetric valleys, an immediate question is how we can find such solutions in practice. It turns out that simply averaging the weights along the SGD trajectory, naturally leads to the desired solutions with bias. We give a theoretical analysis to support this argument, see Figure 1(c) for an illustration. Note that our result is in line with the empirical observations recently made by Izmailov et al. (2018).

In addition, we provide empirical analysis to verify our theoretical results and support our claims. For example, we show that asymmetric valleys are indeed prevalent in modern deep networks, and solutions with lower generalization error has bias towards the flat side of the valley. We also find that batch normalization seems to be a major cause for shaping asymmetric loss surfaces.

2. Related Work

Neural network landscape. Analyzing the landscape of deep neural networks is an active and exciting area (Goodfellow & Vinyals, 2014; Li et al., 2018; Ge et al., 2017; Pennington & Bahri, 2017; Wu et al., 2017; Cooper, 2018; Sagun et al., 2017). For example, (Draxler et al., 2018; Garipov et al., 2018) observed that essentially all local minima are connected together with simple paths. (Huang et al., 2017) used cyclic learning rate and took the ensemble of intermediate models to get improved accuracy. There are also appealing visualizations for the neural network landscape (Li et al., 2018).

Sharp and flat minima. The discussion of sharp and flat local minima dates back to (Hochreiter & Schmidhuber, 1995), and recently regains its popularity. For example, Keskar et al. (2017) proposed that large batch SGD finds sharp minima, which leads to poor generalization. In (Chaudhari et al., 2016), an entropy regularized SGD was introduced to explicitly searching for flat minima. It was later pointed out that large batch SGD can yield comparable performance when

the learning rate or the number of training iterations are properly set (Hoffer et al., 2017; Goyal et al., 2017; Smith et al., 2017; Masters & Luschi, 2018; Smith & Le, 2017; Jastrzebski et al., 2017). Moreover, (Dinh et al., 2017) showed that from a given flat minimum, one could construct another minimum with arbitrarily sharp directions but equally good performance. In this paper, we argue that the description of sharp or flat minima is an oversimplification. There may simultaneously exist steep directions, flat directions, and asymmetric directions for the same minimum.

SGD optimization and generalization. As the de facto optimization tool for deep networks, SGD and its variants are extensively studied in the literature. For example, it is shown that they could escape saddle points or sharp local minima under reasonable assumptions (Ge et al., 2015; Jin et al., 2017; 2018a;b; Xu et al., 2018; Allen-Zhu, 2018a;b; Allen-Zhu & Li, 2018; Kleinberg et al., 2018). For convex functions (Polyak & Juditsky, 1992) or strongly convex but non-smooth functions (Rakhlin et al., 2012), SGD averaging is shown to give better convergence rate. In addition, it can also achieve higher generalization performance for Lipschitz functions in theory (Shalev-Shwartz et al., 2009; Cesa-bianchi et al., 2002), or for deep networks in practice (Huang et al., 2017; Izmailov et al., 2018; ?). Discussions on the generalization bound of neural networks can be found in (Bartlett et al., 2017; Neyshabur et al., 2018; 2017b; Kawaguchi et al., 2017; Neyshabur et al., 2017a; Arora et al., 2018; Zhou et al., 2019).

We show that SGD averaging has implicit bias on the flat sides of the minima. Previously, it was shown that SGD has other kinds of implicit bias as well (Soudry et al., 2017; Ji & Telgarsky, 2018; Gunasekar et al., 2018).

3. Asymmetric Valleys

In this section, we give a formal definition of asymmetric valley, and show that it is prevalent in the loss landscape of modern deep neural networks.

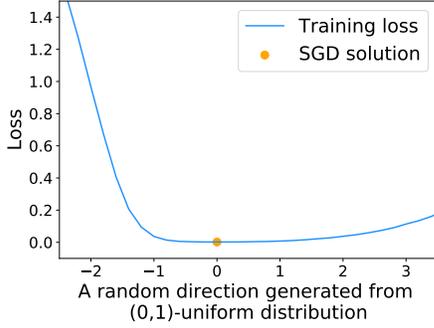


Figure 2. An asymmetric direction of a local minimum on the loss landscape of ResNet-110 trained on CIFAR-10.

Preliminaries. In supervised learning, we seek to optimize $\mathbf{w}^* \triangleq \arg \min_{\mathbf{w} \in \mathbb{R}^d} \mathbb{L}(\mathbf{w})$, where $\mathbb{L}(\mathbf{w}) \triangleq \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[f(\mathbf{x}; \mathbf{w})] \in \mathbb{R}^d \rightarrow \mathbb{R}$ is the population loss, $\mathbf{x} \in \mathbb{R}^m$ is the input from distribution \mathcal{D} , $\mathbf{w} \in \mathbb{R}^d$ denotes the model parameter, and $f \in \mathbb{R}^m \times \mathbb{R}^d \rightarrow \mathbb{R}$ is the loss function.

Since the data distribution \mathcal{D} is usually unknown, instead of optimizing \mathbb{L} directly, we often use SGD to find the empirical risk minimizer $\hat{\mathbf{w}}^*$ for a set of random samples $\{\mathbf{x}_i\}_{i=1}^n$ from \mathcal{D} (a.k.a. training set): $\hat{\mathbf{w}}^* \triangleq \arg \min_{\mathbf{w} \in \mathbb{R}^d} \hat{\mathbb{L}}(\mathbf{w})$, where $\hat{\mathbb{L}}(\mathbf{w}) \triangleq \frac{1}{n} \sum_{i=1}^n f(\mathbf{x}_i; \mathbf{w})$.

We use a unit vector $\mathbf{u} \in \mathbb{R}^d$ to represent a direction such that the points on this direction passing $\mathbf{w} \in \mathbb{R}^d$ can be written as $\mathbf{w} + l\mathbf{u}$ for $l \in (-\infty, \infty)$.

3.1. Definition of asymmetric valley

Before formally introducing asymmetric valleys, we first define asymmetric directions.

Definition 1 (Asymmetric direction). *Given constants $p > 0, r > \zeta > 0, c > 1$, a direction \mathbf{u} is (r, p, c, ζ) -asymmetric with respect to point $\mathbf{w} \in \mathbb{R}^d$ and loss function $\hat{\mathbb{L}}$, if $\nabla_l \hat{\mathbb{L}}(\mathbf{w} + l\mathbf{u}) < p$, and $\nabla_l \hat{\mathbb{L}}(\mathbf{w} - l\mathbf{u}) < -cp$ for $l \in (\zeta, r)$.*

To put it simply, asymmetric direction is a direction \mathbf{u} along which the loss function grows at different rates at the positive/negative direction. The constant ζ handles the small neighborhood around \mathbf{w} with very small gradients. With this definition, we now formally define the *asymmetric valley*.

Definition 2 (Asymmetric valley). *Given constants $p, r > \zeta > 0, c > 1$, a local minimum $\hat{\mathbf{w}}^*$ of $\hat{\mathbb{L}} \in \mathbb{R}^d \rightarrow \mathbb{R}$ is a (r, p, c, ζ) -asymmetric valley, if there exists at least one direction \mathbf{u} such that \mathbf{u} is (r, p, c, ζ) -asymmetric with respect to $\hat{\mathbf{w}}^*$ and $\hat{\mathbb{L}}$.*

Notice that here we abuse the name “valley”, since $\hat{\mathbf{w}}^*$ is essentially a point at the center of a valley.

3.2. Find asymmetric directions empirically

Empirically, by taking random directions with value $(0, 1)$ in each dimension, we could find an asymmetric direction for a given local minimum with decent probability¹. We perform experiments with three widely used deep networks, i.e., ResNet-110, ResNet-164 (He et al., 2016), DenseNet-100 (Huang et al., 2016), on the CIFAR-10 and CIFAR-100 image classification datasets. For each model on each dataset, we conduct 5 independent runs. The results show that we could *always* find asymmetric directions with certain specification (r, p, c, ζ) with $c > 2$, which means all the local minima² being found are located in asymmetric valleys. Figure 2 shows an asymmetric direction for a local minimum in ResNet-110 trained on the CIFAR-10 dataset. We verified that it is a $(2.5, 0.2, 7.5, 1.2)$ -asymmetric direction. Asymmetric valleys widely exist in other models as well, see Appendix A.

4. Bias and Generalization

As we show in the previous section, most local minima in practice are *asymmetric*, i.e., they might be sharp on one direction, but flat on the opposite direction. Therefore, it is important to investigate the generalization ability of a solution \mathbf{w} in this scenario. In this section, we prove that a *biased* solution on the flat side of an asymmetric valley yields lower generalization error than the empirical minimizer $\hat{\mathbf{w}}^*$ in that valley.

4.1. Theoretical analysis

Before presenting our theorem, we first introduce two mild assumptions. We will show that they empirically hold on modern deep networks in Section 4.2.

The first assumption (Assumption 1) states that there exists a shift between the empirical loss and true population loss. This is a common assumption in the previous works, e.g., (Keskar et al., 2017), but was usually presented in an informal way. Here we define the “shift” in a formal way. Without loss of generality, we will compare the empirical loss $\hat{\mathbb{L}}$ with $\mathbb{L}' \triangleq \mathbb{L} - \min_{\mathbf{w}} \mathbb{L} + \min_{\mathbf{w}} \hat{\mathbb{L}}$ to remove the “vertical difference” between $\hat{\mathbb{L}}$ and \mathbb{L} . Notice that $\min_{\mathbf{w}} \mathbb{L}$ and $\min_{\mathbf{w}} \hat{\mathbb{L}}$ are constants and do not affect our generalization guarantee.

Definition 3 ((δ, R) -shift gap). *For $\xi \geq 0, \delta \in \mathbb{R}^d$, and fixed functions \mathbb{L} and $\hat{\mathbb{L}}$, we define the (δ, R) -shift gap between \mathbb{L} and $\hat{\mathbb{L}}$ with respect to a point \mathbf{w} as*

$$\xi_{\delta}(\mathbf{w}) = \max_{\mathbf{v} \in \mathbb{B}(R)} |\mathbb{L}'(\mathbf{w} + \mathbf{v} + \delta) - \hat{\mathbb{L}}(\mathbf{w} + \mathbf{v})|$$

¹By contrast, a random direction with value in $(-1, 1)$ is usually not asymmetric.

²Notice that empirically we could not verify whether the SGD solution $\hat{\mathbf{w}}^*$ is a local minimum. See the discussion in Section 6.

where $L'(\mathbf{w}) \triangleq L(\mathbf{w}) - \min_{\mathbf{w}} L(\mathbf{w}) + \min_{\mathbf{w}} \hat{L}(\mathbf{w})$, and $\mathbb{B}(R)$ is the d -dimensional ball with radius R centered at $\mathbf{0}$.

From the above definition, we know that the two functions match well after the shift δ if $\xi_\delta(\mathbf{w})$ is very small. For example, $\xi_\delta(\mathbf{w}) = 0$ means L is locally identical to \hat{L} after the shift δ . Since \hat{L} is computed on a set of random samples from \mathcal{D} , the actual shift δ between \hat{L} and L is a random variable, ideally with zero expectation.

Assumption 1 (Random shift assumption). *For a given population loss L and a random empirical loss \hat{L} , constants $R > 0, r \geq \zeta > 0, \xi \geq 0$, a vector $\bar{\delta} \in \mathbb{R}^d$ with $r \geq \bar{\delta}_i \geq \zeta$ for all $i \in [d]$, a minimizer $\hat{\mathbf{w}}^*$, we assume that there exists a random variable $\delta \in \mathbb{R}^d$ correlated with \hat{L} such that $\Pr(\delta_i = \bar{\delta}_i) = \Pr(\delta_i = -\bar{\delta}_i) = \frac{1}{2}$ for all $i \in [d]$, and the (δ, R) -shift gap between L and \hat{L} with respect to $\hat{\mathbf{w}}^*$ is bounded by ξ .*

Roughly, the above assumption says that the local landscape of the empirical loss and population loss match well after applying a shift vector δ , which has equal probability of being positive or negative in each dimension. Therefore, δ has 2^d possible values for a given shift vector $\bar{\delta}$, each with probability 2^{-d} . The second assumption stated below can be seen as an extension of Definition 2.

Assumption 2 (Locally asymmetric). *For a given population loss \hat{L} , and a minimizer $\hat{\mathbf{w}}^*$, there exist orthogonal directions $\mathbf{u}^1, \dots, \mathbf{u}^k \in \mathbb{R}^d$ s.t. \mathbf{u}^i is (r, p_i, c_i, ζ) -asymmetric with respect to $\hat{\mathbf{w}}^* + \mathbf{v} - \langle \mathbf{v}, \mathbf{u}^i \rangle \mathbf{u}^i$ for all $\mathbf{v} \in \mathbb{B}(R')$ and $i \in [k]$.*

Assumption 2 states that if \mathbf{u}^i is an asymmetric direction at $\hat{\mathbf{w}}^*$, then the point $\hat{\mathbf{w}}^* + \mathbf{v} - \langle \mathbf{v}, \mathbf{u}^i \rangle \mathbf{u}^i$ that deviates from $\hat{\mathbf{w}}^*$ along the perpendicular direction of \mathbf{u}^i , is also asymmetric along the direction of \mathbf{u}^i . In other words, the neighborhood around $\hat{\mathbf{w}}^*$ is an asymmetric valley.

Under the above assumptions, we are ready to state our theorem, which says the empirical minimizer is not necessarily the optimal solution, while a biased solution leads to better generalization. We defer the proof to Appendix B.

Theorem 1 (Bias leads to better generalization). *For any $\mathbf{l} \in \mathbb{R}^k$, if Assumption 1 holds for $R = \|\mathbf{l}\|_2$, Assumption 2 holds for $R' = \|\bar{\delta}\|_2 + \|\mathbf{l}\|_2$, and $\frac{4\xi}{(c_i-1)p_i} < l_i \leq \max\{r - \bar{\delta}_i, \bar{\delta}_i - \zeta\}$, then we have*

$$\begin{aligned} & \mathbb{E}_\delta L(\hat{\mathbf{w}}^*) - \mathbb{E}_\delta L\left(\hat{\mathbf{w}}^* + \sum_{i=1}^k l_i \mathbf{u}^i\right) \\ & \geq \sum_{i=1}^k (c_i - 1) l_i p_i / 2 - 2k\xi > 0 \end{aligned}$$

Remark on Theorem 1. It is widely known that the empirical minimizer is usually different from the true optimum.

However, in practice it is difficult to know how the training loss shifts from the population loss. Therefore, the best we could do is minimizing the empirical loss function (with some regularizers). On the contrary, Theorem 1 states that under the asymmetric case, we should pick a biased solution to minimize the expected population loss even the shift is unknown. Moreover, it is possible to distill our insight into practical algorithms, as we will discuss in Section 5.

4.2. Verification of assumptions

Verification of Assumption 1. We show that a shift between L and \hat{L} is quite common in practice, by taking a ResNet-110 trained on CIFAR-10 as an example. Since we could not visualize a shift in a high dimensional space, we randomly sample an asymmetric direction \mathbf{u} (more results are shown Appendix C) at the SGD solution $\hat{\mathbf{w}}^*$. The blue and red curves shown in Figure 3(a) are obtained by calculating $\hat{L}(\hat{\mathbf{w}}^* + l\mathbf{u})$ and $L'(\hat{\mathbf{w}}^* + l\mathbf{u})$ for $l \in [-3, 3]$, which correspond to the training and test loss, respectively.

We then try different shift values of δ to “match” the two curves. As shown in Figure 3(a), after applying a horizontal shift $\delta = 0.4$ to the test loss, the two curves overlap almost perfectly. Quantitatively, we can use the *shift gap* defined in Definition 3 to evaluate how well the two curves match each other after shifting. It turns out that $\xi_{\delta=0.4} = 0.0335$, which is much lower than $\xi_{\delta=0} = 0.223$ before shifting (δ has only one dimension here). In Figure 3(b), we plot ξ_δ/ξ_0 as a function of δ . Clearly, there exists a δ that minimizes this ratio, indicating a good match.

We conducted the same experiments for different directions, models and datasets, and similar observations were made. Please refer to Appendix C for more results.

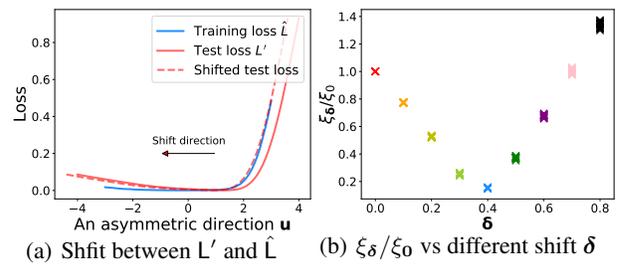


Figure 3. Shift exists between empirical loss and population loss for ResNet-110 on CIFAR-10.

Verification of Assumption 2. This is a mild assumption that can be verified empirically. For example, we take a SGD solution of ResNet-110 on CIFAR-10 as $\hat{\mathbf{w}}^*$, and specify an asymmetric direction \mathbf{u} for $\hat{\mathbf{w}}^*$. We then randomly sample 100 different local adjustments for $\mathbf{v} \in \mathbb{B}(25)$. Based on these adjustments, we present the mean loss curves and standard variance zone on the asymmetric direction \mathbf{u} for

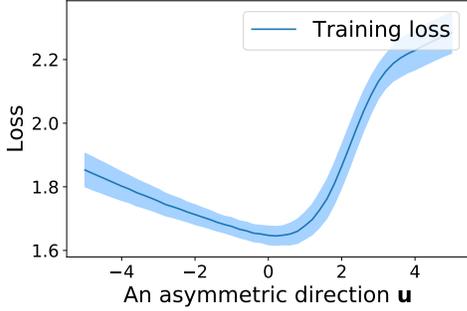


Figure 4. Training loss mean and standard variance for the neighborhood of \hat{w}^* at the direction of \mathbf{u} .

all the points $\hat{w}^* + \mathbf{v} - \langle \mathbf{v}, \mathbf{u} \rangle \mathbf{u}$ in Figure 4. As we can see, the variance of these curves are very small, which means all of them are similar to each other. Moreover, we verified that \mathbf{u} is $(4, 0.1, 5.22, 2)$ -asymmetric with respect to all neighboring points.

5. Averaging Generates Good Bias

In the previous section, we show that when the loss landscape of a local minimum is asymmetric, a solution with bias towards the flat side of the valley has better generalization performance. One immediate question is that how can we obtain such a solution via practical algorithms? Below we show that it can be achieved by simply taking the average of SGD iterates during the course of training. We first analyze the one dimensional case in Section 5.1, and then extend the analysis to the high dimensional case in Section 5.2.

5.1. One dimensional case

For asymmetric functions, as long as the learning rate is not too small, SGD will oscillate between the flat side and the sharp side. Below we focus on one round of oscillation, and show that the average of the iterates in each round has a bias on the flat side. Consequently, by aggregating all rounds of oscillation, averaging SGD iterates leads to a bias as well.

For each individual round i , we assume that it starts from the iteration when SGD goes from sharp side to flat side (denoted as w_0^i), and ends at the iteration exactly before the iteration that SGD goes from sharp side to flat side again (denoted as $w_{T_i}^i$). Here T_i denotes the number of iterations in the i -th rounds. The average iterate in the i -th round can be written as $\bar{w} \triangleq \frac{1}{T_i} \sum_{j=0}^{T_i} w_j^i$. For notational simplicity, we will omit the super script i on w_j^i .

The following theorem shows that the expectation of the average has bias on the flat side. To get a formal lower bound on \bar{w} , we consider the asymmetric case where $\zeta = 0$, and also assume lower bounds for the gradients on the function.

Notice that we made little effort to optimize the constants or bounds on the parameters, and we defer the proof to Appendix E.

Theorem 2 (SGD averaging generates a bias). *Assume that a local minimizer $w^* = 0$ is a $(r, a_+, c, 0)$ -asymmetric valley, where $b_- \leq \nabla L(w) \leq a_- < 0$ for $w < 0$, and $0 < b_+ \leq \nabla L(w) \leq a_+$ for $w \geq 0$. Assume $-a_- = ca_+$ for a large constant c , and $\frac{-(b_- - \nu)}{b_+} = c' < \frac{e^{c/3}}{6}$. The SGD updating rule is $w_{t+1} = w_t - \eta(\nabla L(w) + \omega_t)$ where ω_t is the noise and $|\omega_t| < \nu$, and assume $\nu \leq a_+$. Then we have*

$$\mathbb{E}[\bar{w}] > c_0 > 0,$$

where c_0 is a constant that only depends on η, a_+, a_-, b_+, b_- and ν .

Theorem 2 can be intuitively explained by Figure 5. If we run SGD on this one dimensional function, it will stay at the flat side for more iterations as the magnitude of the gradient on this side is much smaller. Therefore, the average of the locations is biased towards the flat side.



Figure 5. SGD iterates and their average on an asymmetric function: the oscillation case.

Of course, if the learning rate is sufficiently small, there will be no oscillations on the SGD trajectory, as shown in Figure 6. In this case, the bias on the sharp side tends to be closer to the center compared to the bias on the flat side, as the gradient on the sharp side is much larger than the gradient on the flat side, so SGD converges much faster. In other words, even if there is no oscillation and Theorem 2 does not apply, SGD averaging creates more bias on flat sides than sharp sides in expectation. Thus in all the scenarios, taking average of SGD iterates would be beneficial for asymmetric loss function.

In addition, for symmetric loss functions, averaging SGD iterates may also be helpful in terms of denoising (see Appendix D for concrete examples). Therefore, taking the average of the SGD trajectory may always improve generalization, regardless of whether the loss function is symmetric or not.



Figure 6. SGD iterates and their average on an asymmetric function with small learning rate: starting from the sharp side (Left); and starting from the flat side (Right)

5.2. High dimensional case

For high dimensional functions, the analysis on averaging SGD iterates would be more complicated compared to that given in the previous subsection. However, if we only care about the bias on a specific direction \mathbf{u} , we could directly apply Theorem 2 with one additional assumption. Specifically, if the projections of the loss function onto \mathbf{u} along the SGD trajectory satisfy the assumptions in Theorem 2, i.e., being asymmetric and the gradient on both sides have upper and lower bounds, then the claim of Theorem 2 directly applies. This is because only the gradient along the direction \mathbf{u} will affect the SGD trajectory projected onto \mathbf{u} , and we could safely omit all other directions.

Empirically, we find that this assumption generally holds. For a given SGD solution, we fix a random asymmetric direction $\mathbf{u} \in \mathbb{R}^d$, and sample the loss surface on direction \mathbf{u} that passes the t -th epoch of SGD trajectory (denoted as \mathbf{w}_t), i.e., evaluate $\hat{L}(\mathbf{w}_t + l\mathbf{u})$, for $0 \leq t \leq 200$ and $l \in [-15, 15]$.

As shown in the Figure 7, after the first 40 epochs, the projected loss surfaces becomes relatively stable. Therefore, we could directly apply Theorem 2 to the direction \mathbf{u} .

As we will see in Section 6.1, compared with SGD solutions, SGD averaging indeed creates bias along different asymmetric directions, as predicted by our theory.

6. Sharp and Flat Minima Illusion

In this section, we show that *where* the solution locates at a local minimum basin is very important, which is a refinement of judging the generalization performance by the sharpness/flatness of a local minimum. All of our observations support our theoretical analysis in the previous sections.

First we remark that rigorously testing whether a point is a local minimum, or even close to a local minimum, is extremely hard for deep models, see e.g. (Safran & Shamir, 2017). In fact, the Hessian of most empirical solutions still have plenty of small negative eigenvalues (Chaudhari et al., 2016), so technically they are saddle points. But we choose to ignore these technicalities, and treat all these points as

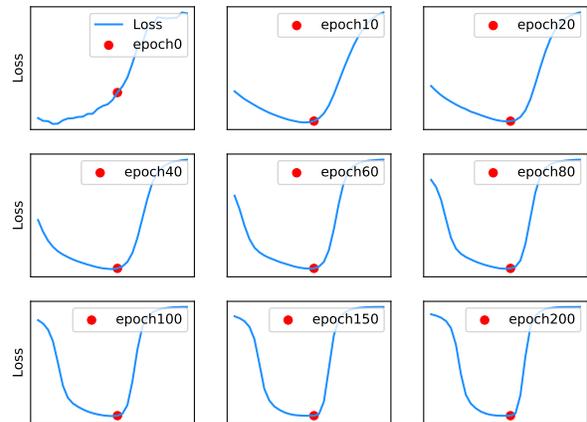


Figure 7. Projection of the training loss surface onto an asymmetric direction \mathbf{u}

“local minima”.

6.1. Illusion case 1: SWA algorithm

Recently, Izmailov et al. (2018) proposed the stochastic weight averaging (SWA) algorithm, which explicitly takes the average of SGD iterates to achieve better generalization. Inspired by their observation that “SWA leads to solutions corresponding to wider optima than SGD”, we provide a more refined explanation in this subsection. That is, averaging weights leads to “biased” solutions in an asymmetric valley, which correspond to better generalization.

Specifically, we run the SWA algorithm (with decreasing learning rate) with three popular deep networks, i.e., ResNet-110, ResNet-164 and DenseNet-100, on the CIFAR-10 and CIFAR-100 datasets, following the configurations in (Izmailov et al., 2018) (denoted as SWA). Then we run SGD with small learning rate *from the SWA solutions* to find a solution located in the same basin (denoted as SGD).

In Figure 8, We draw an interpolation between the solutions obtained by SWA and SGD³. One can observe that there is no “bump” between these two solutions, meaning they are located in the same basin. Clearly, the SWA solution is biased towards the flat side, which verifies our theoretical analysis in Section 5. Further, we notice that although the biased SWA solution has higher training loss than the empirical minimizer, it indeed yields lower test loss. This verifies our analysis in Section 4. Similar observations are made on other networks and other datasets, which we present in Appendix F.

To further support our claim, we list our result in Table 1, from which we can observe that SGD solutions always have higher training accuracy, but worse test accuracy, compared

³Izmailov et al. (2018) have done a similar experiment.

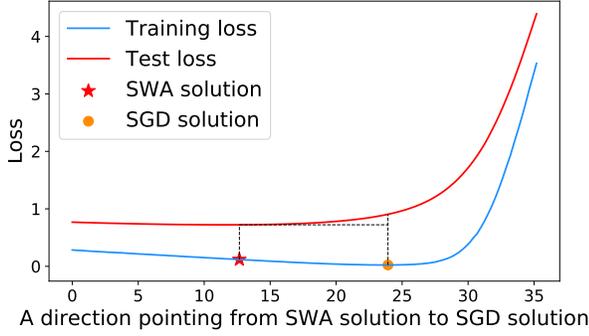


Figure 8. SWA solution and SGD solution interpolation (ResNet-164 on CIFAR100)

to SWA solutions. This supports our claim in Theorem 1, which states that a bias towards the flat sides of asymmetric valleys could help improve generalization, although it yields higher training error.

Table 1. Training accuracy of various networks on CIFAR-100.

Network	CIFAR-100	
	train	test
ResNet-110-SWA	94.98%	78.94%
ResNet-110-SGD	97.52%	78.29%
ResNet-164-SWA	97.48%	80.69%
ResNet-164-SGD	99.12%	76.56%
DenseNet-100-SWA	99.84%	72.29%
DenseNet-100-SGD	99.87%	71.46%

Verifying Theorem 2. We further verify that averaging SGD solutions could create a bias towards the flat side in expectation for many other asymmetric directions, not just for the specific direction we discussed above.

We take a ResNet-110 trained on CIFAR-100 as an example. Denote \mathbf{u}_{inter} as the unit vector pointing from the SGD solution to the SWA solution. We pick another unit random direction \mathbf{u}_{rand} . Then, we use the direction $\mathbf{u}_{inter} + \mathbf{u}_{rand}$ to verify our claim.

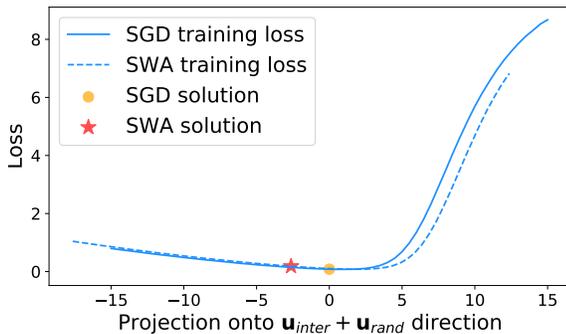


Figure 9. The average of SGD has a bias on flat side (ResNet-110 on CIFAR100)

The results are shown in Figure 9, from which we can observe that SWA has a bias on the flat side compared with the SGD solution. We create 10 different random vectors for each network and each dataset, and similar observations can be made (see more examples in Appendix G).

6.2. Illusion case 2: large batch SGD

Keskar et al. (2017) observed that training with small batch size using SGD algorithm generalizes better than training with large batch size. They argue that it is because large batch SGD tends to converge to sharp minima, while small batch SGD generally converges to flat minima. Here we show that it may not be the case in practice.

We use a PreResNet-164 trained on CIFAR-100 as an example. We first running SGD with a batch size of 128 for 200 epochs to find a solution (denoted as *Large batch solution*), and then continue the training with batch size 32 for another 80 epoch to find a nearby solution (denoted as *Small batch solution*).

From the results shown in Figure 10, it is clear that the small batch solution has worse training accuracy but better test accuracy. Meanwhile, there is no ‘bump’ between these solutions which suggests they are in the same basin. Therefore, small batch SGD generalizes better because it could find a better biased solution in the asymmetric valley, not because it finds a different wider or flatter minimum.

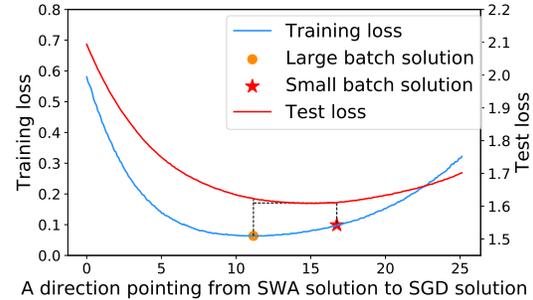


Figure 10. Large and small minibatch interpolation (batch size 128 to 32 of PreResNet-164 on CIFAR-100)

6.3. Illusion on the width of a minimum

We further point out that visualizing the ‘width’ of a local minimum in a low-dimensional space may lead to illusive results. For example, one visualization technique (Izmailov et al., 2018) is showing how the loss changes along many random directions \mathbf{v}_i ’s drawn from the d -dimensional Gaussian distribution.

We take the large batch and small batch solutions from the previous subsection as our example. Figure 11 visualizes the ‘width’ of the two solutions using the method described above. From the figure, one may draw the conclusion that

small batch training leads to a wider minimum compared to large batch training. However, as discussed in Subsection 6.2 these two solutions are actually from the same basin. In other words, the loss curvature near the two solutions looks different because they are located at different locations in an asymmetric valley, instead of being located at different local minima. Similar observation holds for SWA and SGD solutions, see Appendix H.

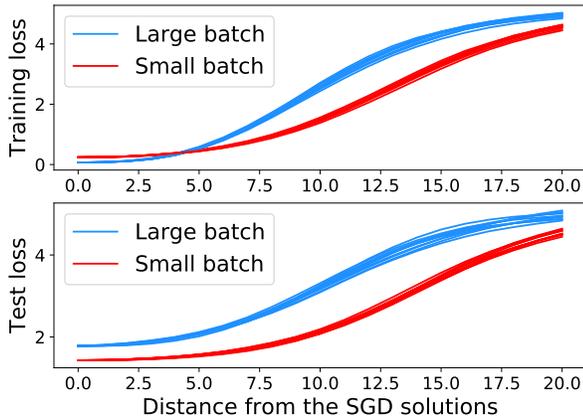


Figure 11. Random ray of large batch and small batch solution (ResNet-164 on CIFAR-100).

7. Batch Norm and Asymmetric Valleys

Previous sections have focused on defining *what* are asymmetric valleys, and *how* to leverage them for better generalization. In this section, we take a step forward to answer *where* they originate, by showing empirical evidences that the Batch Normalization (BN) (Ioffe & Szegedy, 2015) adopted by modern neural networks seems to be a major cause for asymmetric valleys.

Directions on BN parameters are more asymmetric.

For a given SGD solution, if we take a random direction where only the BN parameters have non-zero entries, and compare it with a random direction where only the non-BN parameters have non-zero entries, we observe that those BN-related directions are usually more asymmetric. The result with ResNet-110 on CIFAR-10 is shown in Figure 12. As we can see, the Non-BN direction is sharp on both sides, but BN direction is flat on one side, and sharp on the other side. We also conducted trials with different networks and datasets, and obtained similar results (see Appendix I).

SGD averaging is more effective on BN parameters.

By Theorem 1 and 2, we know that SGD averaging could lead to biased solutions on asymmetric directions with better generalization. If BN indeed creates many asymmetric directions, can we improve the model performance by only averaging the weights of BN layers?

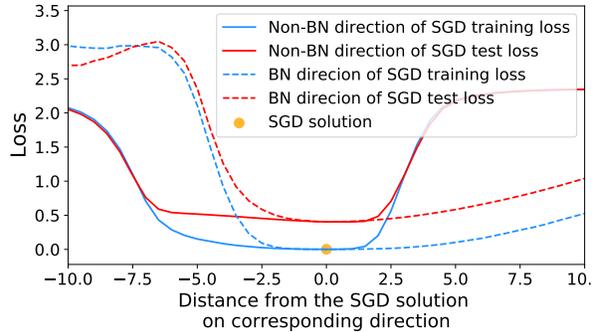


Figure 12. BN and Non-BN directions through a local minimum of ResNet-110 on CIFAR-10.

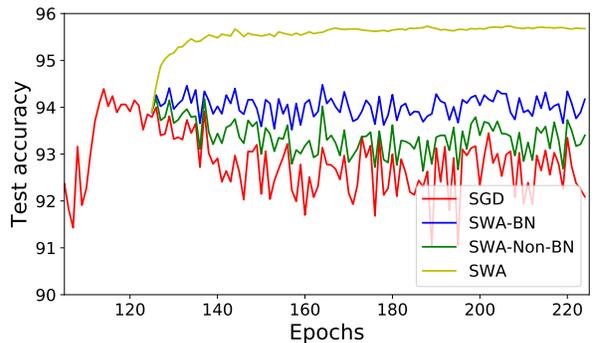


Figure 13. SGD averaging on BN parameters could give better test accuracy compared with SGD averaging on non-BN parameters.

Note that BN parameters only constitute a small fraction of the total model parameters, e.g., 1.41% in a ResNet-110. In the follow experiment on ResNet-110 for CIFAR-10, we perform SGD averaging only on BN parameters, denoted as SWA-BN; and also averaging randomly selected non-BN parameters of the same amount (1.41% of the total parameters), denoted as SWA-Non-BN. The results are shown in Figure 13. It can be observed that averaging only BN parameters (blue curve) is more effective than averaging non-BN parameters (green curve), although there is still a gap comparing to averaging all the weights (yellow curve).

Moreover, we also conduct experiments with two 8-layer ResNets on CIFAR-10, one with BN layers and one without. We choose shallow networks here as deeper models without BN can not be effectively trained.

As shown in figure 14, we start weight averaging at the 126-th epoch. Although in both networks, we observe an improvement in test accuracy after averaging, it is clear that the network with BN layers have larger improvement compared with the network without BN layers. This again indicates that SGD averaging is more effective on BN parameters.

The results presented above are still quite preliminary. Un-

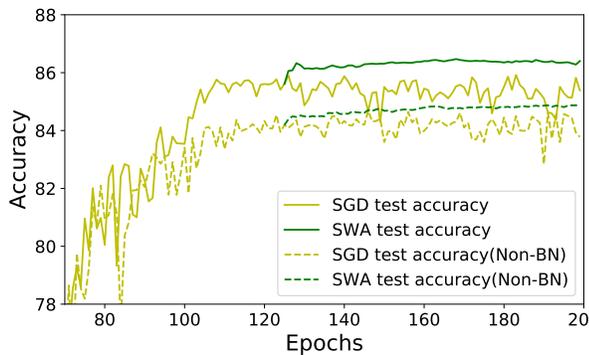


Figure 14. Test accuracy of ResNet-8 with and without BN layers, after running weight averaging (SWA).

derstanding how the asymmetric valleys are formed in deep networks might be a valuable future research direction.

8. Conclusion

The width of solutions has been used to explain generalization. In this paper, we elaborate on these arguments, and show that width along *Asymmetric Valleys*, where the loss may increase at different rates along two opposition directions, is especially important for explaining generalization. Based on a formal definition of asymmetric valley, we showed that a biased solution lying on the flat side of the valley generalizes better than the empirical minimizer. Further, it is proved that by averaging the points along the SGD trajectory naturally leads to such biased solution. We have conducted extensive experiments with state-of-the-art deep models to verify our theorems. We hope this paper will strengthen our understanding on the loss landscape of deep neural networks, and inspire new theories and algorithms that further improve generalization.

References

- Allen-Zhu, Z. How to make the gradients small stochastically. *CoRR*, abs/1801.02982, 2018a.
- Allen-Zhu, Z. Natasha 2: Faster non-convex optimization than sgd. In *Advances in Neural Information Processing Systems 31*, pp. 2680–2691, 2018b.
- Allen-Zhu, Z. and Li, Y. Neon2: Finding local minima via first-order oracles. In *Advances in Neural Information Processing Systems 31*, pp. 3720–3730, 2018.
- Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*, 2018.
- Bartlett, P. L., Foster, D. J., and Telgarsky, M. Spectrally-normalized margin bounds for neural networks. *CoRR*, abs/1706.08498, 2017.
- Cesa-bianchi, N., Conconi, A., and Gentile, C. On the generalization ability of on-line learning algorithms. In *Advances in Neural Information Processing Systems 14*, pp. 359–366. MIT Press, 2002.
- Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-SGD: Biasing Gradient Descent Into Wide Valleys. *ArXiv e-prints*, November 2016.
- Choromanska, A., LeCun, Y., and Arous, G. B. Open problem: The landscape of the loss surfaces of multilayer networks. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pp. 1756–1760, 2015.
- Cooper, Y. The loss landscape of overparameterized neural networks. *CoRR*, abs/1804.10200, 2018.
- Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp Minima Can Generalize For Deep Nets. *ArXiv e-prints*, March 2017.
- Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. Essentially no barriers in neural network energy landscape. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1309–1318, 10–15 Jul 2018.
- Garipov, T., Izmailov, P., Podoprikin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. *CoRR*, abs/1802.10026, 2018.
- Ge, R., Huang, F., Jin, C., and Yuan, Y. Escaping from saddle points - online stochastic gradient for tensor decomposition. In *COLT 2015*, volume 40, pp. 797–842, 2015.
- Ge, R., Lee, J. D., and Ma, T. Learning One-hidden-layer Neural Networks with Landscape Design. *ArXiv e-prints*, November 2017.
- Goodfellow, I. J. and Vinyals, O. Qualitatively characterizing neural network optimization problems. *CoRR*, abs/1412.6544, 2014.
- Goyal, P., Dollár, P., Girshick, R. B., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017.
- Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *PMLR*, volume 80, pp. 1832–1841, 10–15 Jul 2018.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

- Hochreiter, S. and Schmidhuber, J. Simplifying neural nets by discovering flat minima. In *Advances in Neural Information Processing Systems 7*, pp. 529–536. MIT Press, 1995.
- Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems 30*, pp. 1729–1739. Curran Associates, Inc., 2017.
- Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. Densely Connected Convolutional Networks. *ArXiv e-prints*, August 2016.
- Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. Snapshot ensembles: Train 1, get m for free. In *ICLR 2017*, 2017.
- Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 448–456, 2015.
- Izmailov, P., Podoprikin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *Uncertainty in Artificial Intelligence (UAI)*, 2018.
- Jastrzebski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. J. Three factors influencing minima in SGD. *CoRR*, abs/1711.04623, 2017.
- Ji, Z. and Telgarsky, M. Risk and parameter convergence of logistic regression. *CoRR*, abs/1803.07300, 2018.
- Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. *CoRR*, abs/1703.00887, 2017.
- Jin, C., Liu, L. T., Ge, R., and Jordan, M. I. On the local minima of the empirical risk. In *NeurIPS*, pp. 4901–4910, 2018a.
- Jin, C., Netrapalli, P., and Jordan, M. I. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018.*, pp. 1042–1085, 2018b.
- Kawaguchi, K., Pack Kaelbling, L., and Bengio, Y. Generalization in Deep Learning. *ArXiv e-prints*, October 2017.
- Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR 2017*, 2017.
- Kleinberg, R., Li, Y., and Yuan, Y. An Alternative View: When Does SGD Escape Local Minima? *ArXiv e-prints*, February 2018.
- Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems 31*, pp. 6391–6401. Curran Associates, Inc., 2018.
- Masters, D. and Luschi, C. Revisiting small batch training for deep neural networks. *CoRR*, abs/1804.07612, 2018.
- Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *CoRR*, abs/1707.09564, 2017a.
- Neyshabur, B., Bhojanapalli, S., Mcallester, D., and Srebro, N. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems 30*, pp. 5947–5956. Curran Associates, Inc., 2017b.
- Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. Towards understanding the role of over-parametrization in generalization of neural networks. *CoRR*, abs/1805.12076, 2018.
- Pennington, J. and Bahri, Y. Geometry of neural network loss surfaces via random matrix theory. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 2798–2806. PMLR, 2017.
- Polyak, B. T. and Juditsky, A. B. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, July 1992. ISSN 0363-0129.
- Rakhlin, A., Shamir, O., and Sridharan, K. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, 2012.
- Safran, I. and Shamir, O. Spurious Local Minima are Common in Two-Layer ReLU Neural Networks. *ArXiv e-prints*, December 2017.
- Sagun, L., Evci, U., Güney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *CoRR*, abs/1706.04454, 2017.
- Shalev-Shwartz, S., Shamir, O., Sridharan, K., and Srebro, N. Stochastic convex optimization. In *Proceedings of The 22nd Conference on Learning Theory*, 2009.
- Smith, S. L. and Le, Q. V. A Bayesian Perspective on Generalization and Stochastic Gradient Descent. *ArXiv e-prints*, October 2017.
- Smith, S. L., Kindermans, P., and Le, Q. V. Don’t decay the learning rate, increase the batch size. *CoRR*, abs/1711.00489, 2017.

Soudry, D., Hoffer, E., and Srebro, N. The implicit bias of gradient descent on separable data. *arXiv preprint arXiv:1710.10345*, 2017.

Wu, L., Zhu, Z., et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

Xu, Y., Rong, J., and Yang, T. First-order stochastic algorithms for escaping from saddle points in almost linear time. In *Advances in Neural Information Processing Systems 31*, pp. 5535–5545. Curran Associates, Inc., 2018.

Zhou, W., Veitch, V., Austern, M., Adams, R. P., and Orbanz, P. Non-vacuous generalization bounds at the imagenet scale: a PAC-bayesian compression approach. In *International Conference on Learning Representations*, 2019.

A. Additional Figures for Section 3.2: Asymmetric Directions

See Figure 15, Figure 16, Figure 17, Figure 18, and Figure 19.

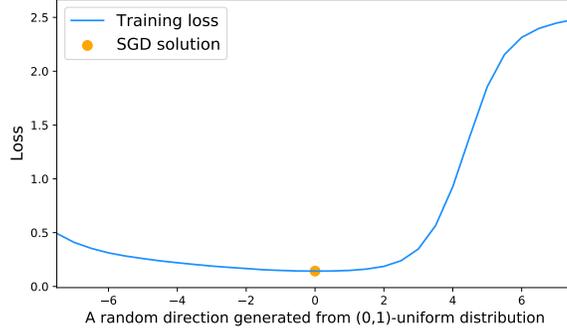


Figure 15. Asymmetric direction for a solution of ResNet-164 on CIFAR-10. $(r, p, c, \zeta) = (4.0, 0.0270, 12.1, 2.0)$.

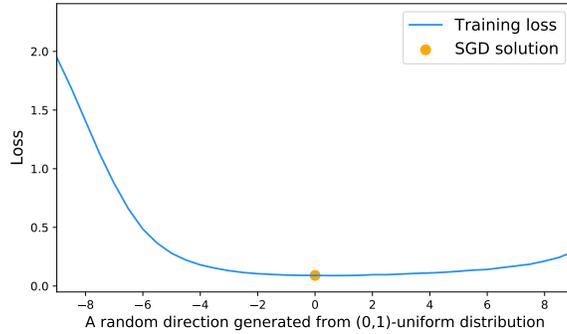


Figure 16. Asymmetric direction for a solution of DenseNet-100 on CIFAR-10. $(r, p, c, \zeta) = (5.0, 0.00022, 452.5, 1.5)$.

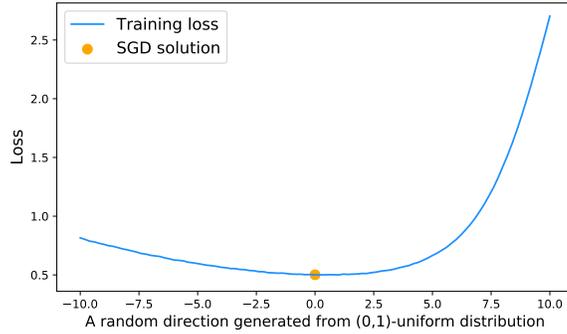


Figure 17. Asymmetric direction for a solution of ResNet-110 on CIFAR-100. $(r, p, c, \zeta) = (8.0, 0.032, 5.52, 4.0)$.

B. Missing Proof for Theorem 1

Proof. Since δ has 2^d possible values for a given $\bar{\delta}$, we can use an integer $j \in \{0, \dots, 2^d - 1\}$ to represent each value. When writing j in binary, its i -th digit represents whether $\delta_i = \bar{\delta}_i$ (equal to 1) or $\delta_i = -\bar{\delta}_i$ (equal to 0). We use $j \wedge 2^i$ to represent the bitwise AND operator between j and 2^i , which equals 0 if the i -th digit of j is 0.

To prove our theorem, it suffices to show that for any $i \in [k]$,

$$\mathbb{E}_{\delta} \mathbb{L} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} l_{i_0} \mathbf{u}_{i_0} \right) - \mathbb{E}_{\delta} \mathbb{L} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^i l_{i_0} \mathbf{u}_{i_0} \right) \geq (c_i - 1) l_i p_i / 2 - 2\xi > 0 \quad (1)$$

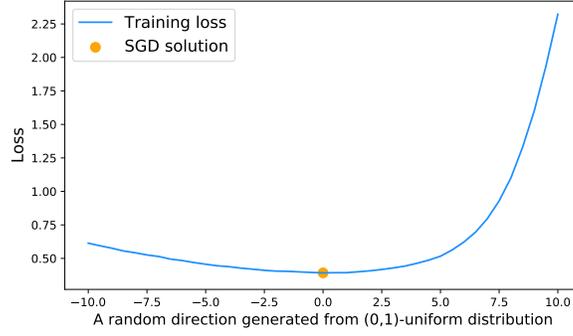


Figure 18. Asymmetric direction for a solution of ResNet-164 on CIFAR-100. $(r, p, c, \zeta) = (7.0, 0.0175, 7.66, 3.0)$.

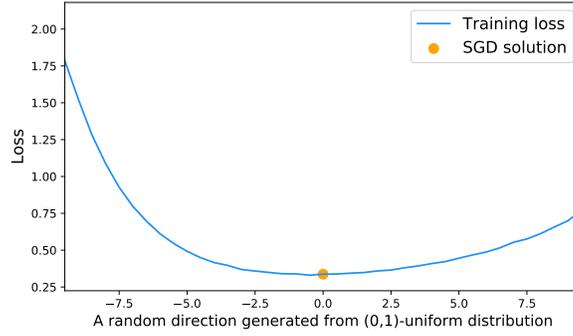


Figure 19. Asymmetric direction for a solution of DenseNet-100 on CIFAR-100. $(r, p, c, \zeta) = (5, 0.0198, 2.259, 2.5)$.

If (1) is true, it suffices to take summation over i on both sides, and we will get our conclusion. Therefore, below we will prove (1).

$$\begin{aligned}
 & \mathbb{E}_{\delta} L \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} \right) - \min_{\mathbf{w}} L(\mathbf{w}) + \min_{\mathbf{w}} \hat{L}(\mathbf{w}) \\
 &= \mathbb{E}_{\delta} L' \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} \right) \stackrel{\textcircled{1}}{\geq} \frac{1}{2^d} \sum_{j=0}^{2^d-1} \hat{L} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \delta^j \right) - \xi \\
 &= \frac{1}{2^d} \sum_{\substack{j=0 \\ j \wedge 2^i = 0}}^{2^d-1} \left[\hat{L} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \delta^j \right) + \hat{L} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \delta^{j+2^i} \right) \right] - \xi
 \end{aligned} \tag{2}$$

Where $\textcircled{1}$ holds by Assumption 1, and the fact that $\| \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} \|_2 \leq \| \mathbf{l} \|_2 = R$. For every j s.t. $j \wedge 2^i = 0$,

$$\begin{aligned}
 & \hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \delta^j \\
 &= \hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \delta^j + \langle \delta^j, \mathbf{u}^i \rangle \mathbf{u}^i - \langle \delta^j, \mathbf{u}^i \rangle \mathbf{u}^i \\
 &= \hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \delta^j - \bar{\delta}_i \mathbf{u}^i - \langle \delta^j, \mathbf{u}^i \rangle \mathbf{u}^i + \mathbf{l}_i \mathbf{u}^i
 \end{aligned}$$

$$= \hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^j - \langle \boldsymbol{\delta}^j, \mathbf{u}^i \rangle \mathbf{u}^i + (\mathbf{l}_i - \bar{\boldsymbol{\delta}}_i) \mathbf{u}^i$$

Since $\|\sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0}\|_2 \leq \|\mathbf{l}\|_2$, $\|\boldsymbol{\delta}^j\|_2 = \|\bar{\boldsymbol{\delta}}\|_2$, we know that $\forall j, \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^j \in \mathbb{B}(R')$. By Assumption 2, for every $i \in [k]$, \mathbf{u}^i is asymmetric with respect to $\hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^j - \langle \boldsymbol{\delta}^j, \mathbf{u}^i \rangle \mathbf{u}^i$. Since $\mathbf{l}_i \leq \bar{\boldsymbol{\delta}}_i - \zeta$, we have $\mathbf{l}_i - \bar{\boldsymbol{\delta}}_i < -\zeta$. By the definition of asymmetric direction, we know

$$\hat{\mathbf{L}} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^j \right) \geq \hat{\mathbf{L}} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^j \right) + c_i \mathbf{l}_i p_i \quad (3)$$

Similarly,

$$\begin{aligned} & \hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^{j+2^i} \\ &= \hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^{j+2^i} + \langle \boldsymbol{\delta}^{j+2^i}, \mathbf{u}^i \rangle \mathbf{u}^i - \langle \boldsymbol{\delta}^{j+2^i}, \mathbf{u}^i \rangle \mathbf{u}^i + \mathbf{l}_i \mathbf{u}^i \\ &= \hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^{j+2^i} - \langle \boldsymbol{\delta}^{j+2^i}, \mathbf{u}^i \rangle \mathbf{u}^i + (\bar{\boldsymbol{\delta}}_i + \mathbf{l}_i) \mathbf{u}^i \end{aligned}$$

Since $\mathbf{l}_i \leq r - \bar{\boldsymbol{\delta}}_i$, we have $\bar{\boldsymbol{\delta}}_i + \mathbf{l}_i \leq r$. Therefore,

$$\hat{\mathbf{L}} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^{i-1} \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^{j+2^i} \right) \geq \hat{\mathbf{L}} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^{j+2^i} \right) - \mathbf{l}_i p_i \quad (4)$$

Combining (3) and (4), we have,

$$\begin{aligned} (2) & \geq \frac{1}{2^d} \sum_{\substack{j=0 \\ j \wedge 2^i = 0}}^{2^d-1} \left[\hat{\mathbf{L}} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^j \right) + c_i \mathbf{l}_i p_i + \hat{\mathbf{L}} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^{j+2^i} \right) - \mathbf{l}_i p_i \right] - \xi \\ &= \frac{1}{2^d} \sum_{j=0}^{2^d-1} \left[\hat{\mathbf{L}} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} + \boldsymbol{\delta}^j \right) \right] + (c_i - 1) \mathbf{l}_i p_i / 2 - \xi \\ &\stackrel{\textcircled{2}}{\geq} \mathbb{E}_\delta \mathbf{L}' \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} \right) + (c_i - 1) \mathbf{l}_i p_i / 2 - 2\xi \\ &= \mathbb{E}_\delta \mathbf{L} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} \right) - \min_{\mathbf{w}} \mathbf{L}(\mathbf{w}) + \min_{\mathbf{w}} \hat{\mathbf{L}}(\mathbf{w}) + (c_i - 1) \mathbf{l}_i p_i / 2 - 2\xi \end{aligned}$$

Where $\textcircled{2}$ holds by Assumption 1 and the fact that $\|\sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0}\|_2 \leq \|\mathbf{l}\|_2 = R$. That means,

$$\mathbb{E}_\delta \mathbf{L} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} \right) \geq \mathbb{E}_\delta \mathbf{L} \left(\hat{\mathbf{w}}^* + \sum_{i_0=1}^i \mathbf{l}_{i_0} \mathbf{u}_{i_0} \right) + (c_i - 1) \mathbf{l}_i p_i / 2 - 2\xi > 0$$

Where the last inequality holds as $\mathbf{l}_i > \frac{4\xi}{(c_i-1)p_i}$.

□

C. Additional Figures for Section 4.2: Shift Exists Empirically

See Figure 20, Figure 21, and Figure 22.

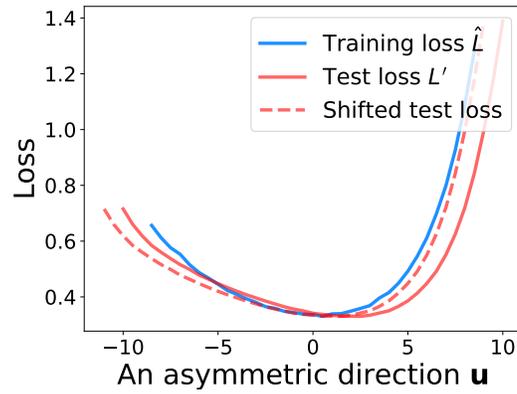


Figure 20. Shift on asymmetric direction (DenseNet-100 on CIFAR-100), $\xi_{\delta=1} = 0.119$, $\xi_{\delta=0} = 0.439$

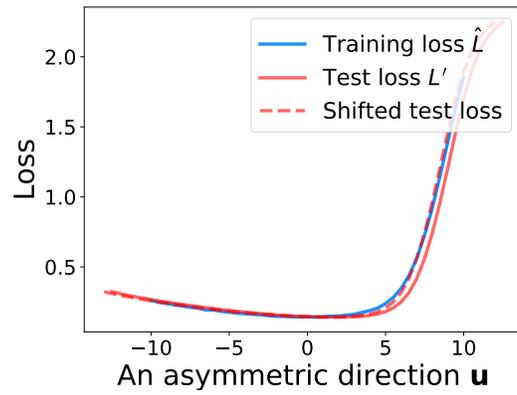


Figure 21. Shift on asymmetric direction (ResNet-164 on CIFAR-10), $\xi_{\delta=0.5} = 0.0699$, $\xi_{\delta=0} = 0.189$

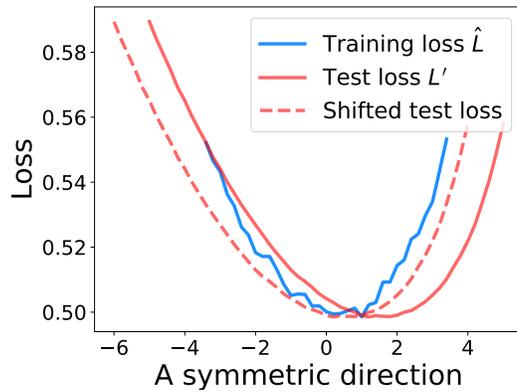


Figure 22. Shift on symmetric direction (ResNet-110 on CIFAR-100), $\xi_{\delta=1} = 0.0197$, $\xi_{\delta=0} = 0.0431$

D. Additional Figures in Section 5: Averaging Works For Symmetric Case

If the function is symmetric, there are two possible cases, as we show in Figure 23 and Figure 24. On one hand, if the function is flat, SGD is likely to stay on one side of the function along the trajectory, and the average will have bias on that side. On the other hand, if the function is sharp, SGD is likely to oscillate between the two sides, and therefore the average of the iterates will concentrate around the center. In both cases, SGD averaging could help to create bias on flat sides or to denoise.

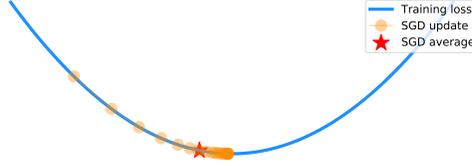


Figure 23. Symmetric function with flat sides

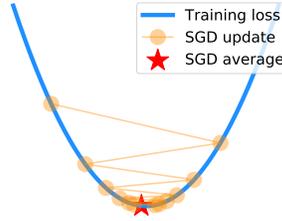


Figure 24. Symmetric function with sharp sides

E. Missing Proof for Theorem 2

To prove Theorem 2, we will need the following concentration bound.

Lemma 3 (Azuma's inequality). *Let $X_1, X_2, X_3, \dots, X_n$ be independent random variables satisfying $|X_i - \mathbb{E}[X_i]| \leq c_i$, for $1 \leq i \leq n$. We have the following bound for $X = \sum_{i=1}^n X_i$:*

$$\Pr(|X - \mathbb{E}(X)| \geq \lambda) \leq 2e^{-\frac{\lambda^2}{2 \sum_{i=1}^n c_i^2}}$$

Let $p_{\min} \triangleq -\eta(a_- + a_+ + 2\nu)$, $p_{\max} \triangleq -\eta(b_- - \nu)$. Since $-a_- = ca_+$, we know $p_{\min} > (c-1)\eta a_+ - 2\eta\nu$. First, we have the following bounds on the first step w_0 .

Lemma 4. *For every $i \in [h]$, $w_0 \in [p_{\min}, p_{\max}]$.*

Proof. Since w_0 is the first step that SGD jumps from the flat side to the sharp side, denote the previous location as $w_{-1} < 0$. Since w_{-1} is at the sharp side, we know that the gradient is $\nabla L(w_{-1}) \leq a_-$. Therefore, we have

$$w_0 = w_{-1} - \eta(\nabla L(w_{-1}) + \omega_{-1})$$

Where ω_{-1} is the noise bounded by ν .

At the time when SGD jump from the flat side to sharp side, denote the target position as w'_{-1} . We know that $w'_{-1} \in [-\eta(a_+ + \nu), 0]$. Since the gradient on the sharp side is at most a_- , we know the next step is lower bounded by $-\eta(a_+ + 2\nu + a_-) = p_{\min} > 0$. In other words, SGD stays at the sharp side for only 1 iterations (this matches with our empirical observation, see e.g. Figure 5).

That means, the bound on w'_{-1} can be applied to w_{-1} as well, because they are the same iterate. By applying the upper and lower bound on $\nabla L(w_{-1})$, we get:

$$w_0 \geq -\eta(a_+ + \nu) - \eta(a_- + \nu) = p_{\min}$$

and also

$$w_0 \leq 0 - \eta(b_- - \nu) = p_{\max}$$

□

Below we first define $T_{\min} \triangleq \left(\frac{-\sqrt{2}\nu \log^{1/2}(2\tau) + \sqrt{2\nu^2 \log(2\tau) - 4a_+(a_- + a_+ + 2\nu)}}{2a_+} \right)^2$, where τ is a constant with value to be set later. T_{\min} satisfies the following inequality.

Lemma 5. $\forall t \leq T_{\min}, p_{\min} - t\eta a_+ - \sqrt{2t}\eta\nu \log^{1/2}(2\tau) \geq 0$.

Proof. By the definition of p_{\min} , we have

$$\begin{aligned} & -\eta(a_- + a_+ + 2\nu) - t\eta a_+ - \sqrt{2t}\eta\nu \log^{1/2}(2\tau) \geq 0 \\ \Leftrightarrow & (a_- + a_+ + 2\nu) + ta_+ + \sqrt{2t}\nu \log^{1/2}(2\tau) \leq 0 \\ \Leftrightarrow & (a_- + a_+ + 2\nu) + \Delta^2 a_+ + \sqrt{2}\Delta r \log^{1/2}(2\tau) \leq 0 \quad (\Delta \triangleq \sqrt{t}) \\ \Leftrightarrow & \Delta \in \left[0, \frac{-\sqrt{2}\nu \log^{1/2}(2\tau) + \sqrt{2\nu^2 \log(2\tau) - 4a_+(a_- + a_+ + 2\nu)}}{2a_+} \right] \\ \Leftrightarrow & t \leq \left(\frac{-\sqrt{2}\nu \log^{1/2}(2\tau) + \sqrt{2\nu^2 \log(2\tau) - 4a_+(a_- + a_+ + 2\nu)}}{2a_+} \right)^2 \end{aligned}$$

□

Now, we have the following theorem that says with decent probability, the minimum number of iterates on the flat side in i -th round is at least T_{\min} .

Theorem 6. *If we start at $w_0 \geq p_{\min}$, for every fixed $\tau > T_{\min}$, with probability at least $1 - \frac{T_{\min}}{\tau}$, we have $\forall t \leq T_{\min}, w_t > w_0 - t\eta a_+ - \sqrt{2t}\eta\nu \log^{1/2}(2\tau) \geq 0$.*

Proof. Define filtration $\mathcal{F}_t = \sigma\{\omega_0, \dots, \omega_{t-1}\}$, where $\sigma\{\cdot\}$ denotes the sigma field. Define the event $\mathfrak{E}_T = \{\forall t \leq T, w_t > w_0 - t\eta a_+ - \sqrt{2t}\eta\nu \log^{1/2}(2\tau)\}$ and define $G_t = w_0 - w_t - t\eta a_+ + M$, where $M \triangleq (T_{\min} + 1)(w_0 + \nu + 2\eta a_+)$. Since we only consider the case $t \leq T_{\min}$, we have

$$G_t = w_0 - w_t - t\eta a_+ + (T_{\min} + 1)(w_0 + \nu + 2\eta a_+) > w_0 - w_t - t\eta a_+ + w_t + t\eta a_+ > 0$$

Therefore, G_t is always positive. By SGD updating rule, we have

$$\begin{aligned} \mathbb{E}[G_{t+1}\mathbb{1}_{\mathfrak{E}_t} | \mathcal{F}_t] &= \mathbb{E}[(w_0 - w_{t+1} - (t+1)\eta a_+ + M)\mathbb{1}_{\mathfrak{E}_t} | \mathcal{F}_t] \\ &\leq \mathbb{E}[(w_0 - w_t + \eta\omega_t - t\eta a_+ + M)\mathbb{1}_{\mathfrak{E}_t} | \mathcal{F}_t] = w_0 - w_t - t\eta a_+ + M = G_t\mathbb{1}_{\mathfrak{E}_t} \end{aligned} \quad (5)$$

Since $\mathbb{1}_{\mathfrak{E}_t} \leq \mathbb{1}_{\mathfrak{E}_{t-1}}$, and G_t is always positive, we have

$$G_t\mathbb{1}_{\mathfrak{E}_t} \leq G_t\mathbb{1}_{\mathfrak{E}_{t-1}} \quad (6)$$

Combining (5) and (6) together, we know $G_t\mathbb{1}_{\mathfrak{E}_{t-1}}$ is a supermartingale.

We can also bound the absolute value of the difference in every iteration:

$$\begin{aligned} & |G_{t+1}\mathbb{1}_{\mathfrak{E}_t} - \mathbb{E}[G_{t+1}\mathbb{1}_{\mathfrak{E}_t} | \mathcal{F}_t]| \\ &= |(w_0 - w_{t+1} - (t+1)\eta a_+ + M) - (w_0 - w_t - \nabla L(w_t) - (t+1)\eta a_+ + M)|\mathbb{1}_{\mathfrak{E}_t}| \\ &\leq \eta\nu \end{aligned}$$

By Azuma's inequality, we get:

$$\Pr(G_t \mathbb{1}_{\mathfrak{E}_{t-1}} - G_0 \geq \lambda) \leq 2e^{-\frac{\lambda^2}{2t\eta^2\nu^2}}$$

That gives,

$$\Pr(G_t \mathbb{1}_{\mathfrak{E}_{t-1}} - G_0 \geq \sqrt{2t\eta\nu} \log^{1/2}(2\tau)) \leq 1/\tau$$

That means, if $\mathbb{1}_{\mathfrak{E}_{t-1}}$ holds, with probability at least $1 - 1/\tau$,

$$w_0 - w_t - t\eta a_+ + M < \sqrt{2t\eta\nu} \log^{1/2}(2\tau) + G_0 = \sqrt{2t\eta\nu} \log^{1/2}(2\tau) + M$$

Which gives

$$w_t > w_0 - t\eta a_+ - \sqrt{2t\eta\nu} \log^{1/2}(2\tau)$$

In other words, that means if $\mathbb{1}_{\mathfrak{E}_{t-1}}$ holds, then $\mathbb{1}_{\mathfrak{E}_t}$ also holds with probability at least $1 - 1/\tau$.

Therefore, if we are running T_{\min} steps, we know that with probability at least $1 - \frac{T_{\min}}{\tau}$, $\mathbb{1}_{\mathfrak{E}_{T_{\min}}}$ holds. Therefore, by Lemma 5,

$$\forall t \leq T_{\min}, w_t > w_0 - t\eta a_+ - \sqrt{2t\eta\nu} \log^{1/2}(2\tau) \geq p_{\min} - t\eta a_+ - \sqrt{2t\eta\nu} \log^{1/2}(2\tau) \geq 0 \quad \square$$

Similarly, we define $T_{\max} \triangleq \left(\frac{-\sqrt{2\nu} \log^{1/2}(2\tau) + \sqrt{2\nu^2 \log(2\tau) - 4(b_- - \nu)b_+}}{2b_+} \right)^2$, which satisfies the following inequality.

Lemma 7. $p_{\max} - T_{\max}\eta b_+ - \sqrt{2T_{\max}\eta\nu} \log^{1/2}(2\tau) < 0$.

Proof. By the definition of p_{\max} , we want to show that

$$(b_- - \nu) + T_{\max}b_+ + \sqrt{2T_{\max}\eta\nu} \log^{1/2}(2\tau) \geq 0$$

Which holds by the definition of T_{\max} . □

The Theorem below shows with decent probability, $T_{\max} - 1$ is an upper bound on the total number of iterates on the flat side in the i -th round.

Theorem 8. *If $w_0 \leq p_{\max}$, with probability at least $1 - \frac{T_{\max}}{\tau}$, $w_{T_{\max}} < 0$.*

Proof. Define event $\mathfrak{E}'_T = \{\forall t \leq T, w_t < w_0 - t\eta b_+ + \sqrt{2t\eta\nu} \log^{1/2}(2\tau)\}$, and $G'_t = w_t + t\eta b_+ > 0$.

We have

$$\begin{aligned} & \mathbb{E}[G'_{t+1} \mathbb{1}_{\mathfrak{E}'_t} | \mathcal{F}_t] \\ &= \mathbb{E}[(w_{t+1} + (t+1)\eta b_+) \mathbb{1}_{\mathfrak{E}'_t} | \mathcal{F}_t] \\ &\leq \mathbb{E}[(w_t - \eta\omega_t + t\eta b_+) \mathbb{1}_{\mathfrak{E}'_t} | \mathcal{F}_t] \\ &= (w_t + t\eta b_+) \mathbb{1}_{\mathfrak{E}'_t} \\ &= G'_t \mathbb{1}_{\mathfrak{E}'_t} \end{aligned}$$

Moreover, we know $\mathbb{1}_{\mathfrak{E}'_t} \leq \mathbb{1}_{\mathfrak{E}'_{t-1}}$, which means $G'_t \mathbb{1}_{\mathfrak{E}'_t} \leq G'_t \mathbb{1}_{\mathfrak{E}'_{t-1}}$. So $G'_t \mathbb{1}_{\mathfrak{E}'_{t-1}}$ is a supermartingale.

We can also bound the absolute value of the difference in every iteration:

$$|G'_{t+1} \mathbb{1}_{\mathfrak{E}'_t} - \mathbb{E}[G'_{t+1} \mathbb{1}_{\mathfrak{E}'_t} | \mathcal{F}_t]|$$

$$\begin{aligned}
 &= |(w_{t+1} + (t+1)\eta b_+) - (w_t - \eta \nabla L(w_t) + (t+1)\eta b_+)|_{\mathcal{F}_t}| \\
 &\leq \eta \nu
 \end{aligned}$$

Using Azuma inequality, we get

$$\Pr \left(G'_t \mathbb{1}_{\mathcal{E}'_{t-1}} - G'_0 \geq \sqrt{2t}\eta\nu \log^{1/2}(2\tau) \right) \leq 2e^{-\frac{t\eta^2\nu^2 \log(2\tau)}{t\eta^2\nu^2}} = \frac{1}{\tau}$$

That means, if $\mathbb{1}_{\mathcal{E}'_{t-1}}$ holds, with probability at least $1 - 1/\tau$,

$$w_t < w_0 - t\eta b_+ + \sqrt{2t}\eta\nu \log^{1/2}(2\tau)$$

In other words, $\mathbb{1}_{\mathcal{E}'_t}$ also holds. Therefore, if we are running T_{\max} steps, we know that with probability at least $1 - \frac{T_{\max}}{\tau}$, $\mathbb{1}_{\mathcal{E}'_{T_{\max}}}$ holds. Therefore, by Lemma 7, we know

$$w_{T_{\max}} < w_0 - T_{\max}\eta b_+ - \sqrt{2T_{\max}}\eta\nu \log^{1/2}(2\tau) < 0 \quad \square$$

Remark. To make sure Theorem 6 is not vacuous, we need to make sure that $T_{\min} \geq 1$. If we want to make T_{\min} , say, at least 2, by Lemma 5, we have:

$$p_{\min} - 2\eta a_+ - 2\eta\nu \log^{1/2}(2\tau) \geq 0$$

Notice that $p_{\min} > (c-1)\eta a_+ - 2\eta\nu$, so we could solve the above inequality and get

$$\begin{aligned}
 &(c-1)\eta a_+ - 2\eta\nu - 2\eta a_+ - 2\eta\nu \log^{1/2}(2\tau) \geq 0 \\
 \Rightarrow &\frac{(c-3)a_+ - 2\nu}{2\nu} \geq \log^{1/2}(2\tau) \\
 \Rightarrow \tau &\leq \frac{e^{\left(\frac{(c-3)a_+ - 1}{2\nu}\right)^2}}{2}
 \end{aligned}$$

Since we assume that c is a large constant and $a_+ \geq \nu$, so τ can be fairly large in order to make sure $T_{\min} \geq 2$. We also know that $T_{\min} \leq \frac{-(a_+ + 2\nu)}{a_+} < c$.

On the other hand, by simple calculation, we know $T_{\max} \leq \frac{-(b_- - \nu)}{b_+} < c' < \frac{e^{c/3}}{6}$. Therefore, we can always pick a τ such that $\frac{T_{\min} + T_{\max}}{\tau} \leq \frac{1}{2}$. So finally, we are ready to prove Theorem 2.

Proof of Theorem 2. By Lemma 4 and Theorem 8, T_{\max} is an upper bound on the length of the i -th round. By Theorem 6, we know that SGD will stay at flat side for at least T_{\min} steps, and each step is lower bounded by $w_t > w_0 - t\eta a_+ - \sqrt{2t}\eta\nu \log^{1/2}(2\tau)$, therefore we know that with probability $1 - \frac{T_{\min} + T_{\max}}{\tau}$:

$$\begin{aligned}
 \frac{1}{T_i} \sum_{j=0}^{T_i} w_j^i &\geq \frac{1}{T_{\max}} \left(\sum_{t=0}^{T_{\min}} [w_0 - t\eta a_+ - \sqrt{2t}\eta\nu \log^{1/2}(2\tau)] - \eta(a_+ + \nu) \right) \\
 &\geq \frac{1}{T_{\max}} \left(\eta a_+ \frac{(T_{\min} + 1)T_{\min}}{2} + \sqrt{2T_{\min}}\eta\nu \log^{1/2}(2\tau) - \eta(a_+ + \nu) \right) \\
 &\geq \frac{T_{\min}^2}{T_{\max}} \eta a_+
 \end{aligned}$$

The above inequality discussed the scenario when Theorem 6 and Theorem 8 hold. If they do not hold, which happens with probability at most $\frac{T_{\min} + T_{\max}}{\tau}$, we need to get lower bound for $\frac{1}{T_i} \sum_{j=0}^{T_i} w_j^i$. Notice that by Lemma 4, we know that SGD

stays at the sharp side for at most 1 iterate in each round, and also the iterates on the flat sides are always positive with $w_0 \geq p_{\min} > \eta(a_+ + \nu)$. Therefore, we have the following trivial bound:

$$\frac{1}{T_i} \sum_{j=0}^{T_i} w_j^i \geq \frac{-\eta(a_+ + \nu) + w_0}{2} > 0$$

Combining two cases together we get

$$\mathbb{E} \left[\frac{1}{T_i} \sum_{j=0}^{T_i} w_j^i \right] \geq \left(1 - \frac{T_{\min} + T_{\max}}{\tau} \right) \frac{T_{\min}^2}{T_{\max}} \eta a_+ + 0$$

Since we can pick τ s.t. $\frac{T_{\min} + T_{\max}}{\tau} \leq \frac{1}{2}$, we have

$$\mathbb{E} \left[\frac{1}{T_i} \sum_{j=0}^{T_i} w_j^i \right] \geq \frac{T_{\min}^2}{2T_{\max}} \eta a_+ \triangleq c_0 > 0$$

□

F. Additional Figures in Section 6.1: No Bumps Between SGD and SWA Solutions

Asymmetric valley of ResNet-110 on CIFAR-10, $(r, p, c, \zeta) = (5, 0.005, 25, 3)$. See Figure 25.

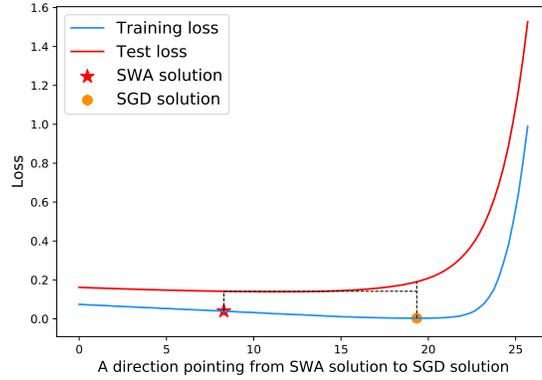


Figure 25. SWA and SGD interpolation (ResNet-110 on CIFAR-10)

Asymmetric valley of ResNet-164 on CIFAR-10, $(r, p, c, \zeta) = (2, 0.015, 6, 1)$. See Figure 26.

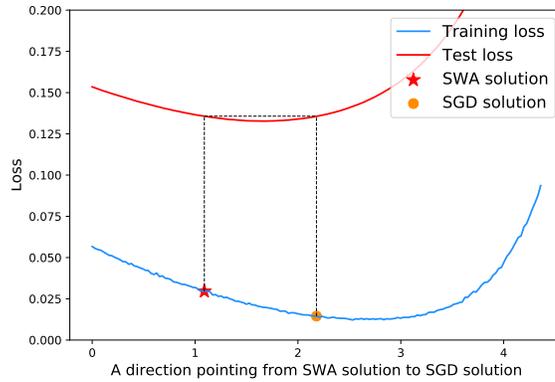


Figure 26. SWA and SGD interpolation (ResNet-164 on CIFAR-10)

Asymmetric valley of DenseNet-100 on CIFAR-10, $(r, p, c, \zeta) = (6, 7.35e - 05, 699, 3)$. See Figure 27

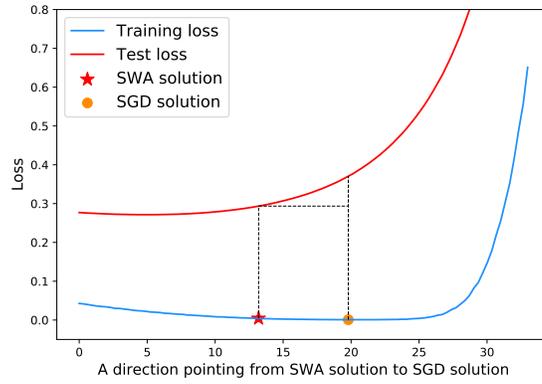


Figure 27. SWA and SGD interpolation (DenseNet-100 on CIFAR-10)

G. Additional Figures in Section 6.1: SGD Averaging Generates Good Bias

Examples for asymmetric directions of ResNet-110 on CIFAR-100 in Figure 28.

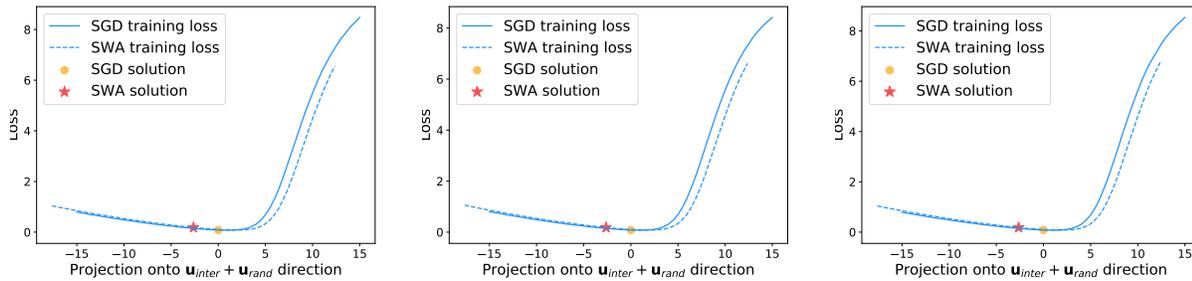


Figure 28. The average of SGD has a bias on flat side (ResNet-110 on CIFAR-100).

Examples for asymmetric directions of ResNet-164 on CIFAR-100 in Figure 29,

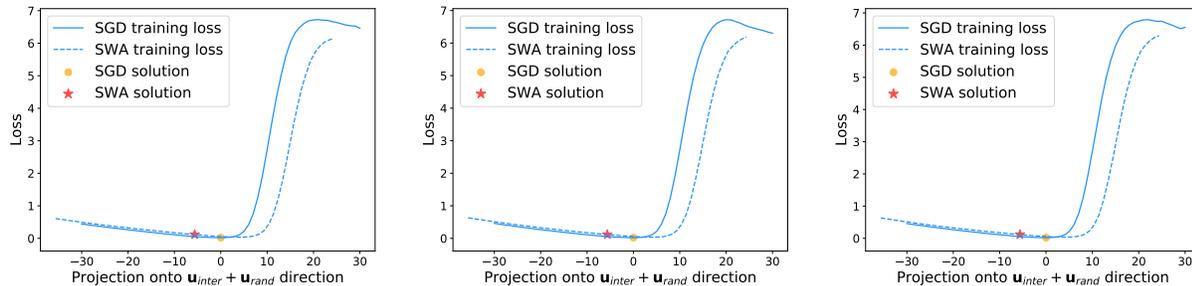


Figure 29. The average of SGD has a bias on flat side (ResNet-164 on CIFAR-100).

Examples for asymmetric directions of ResNet-110 on CIFAR-10 in Figure 30.

H. Additional Figure for Section 6.3: Width of Minima

See Figure 31, similar results were observed by (Izmailov et al., 2018) as well.

I. Additional Figures for Section 7: BN Parameters Are More Asymmetric

See Figure 32, Figure 33 and Figure 34.

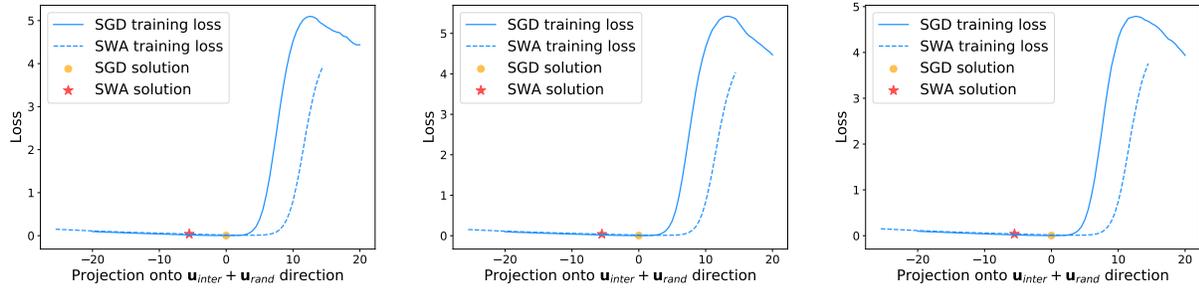


Figure 30. The average of SGD has a bias on flat side (ResNet-110 on CIFAR-10).

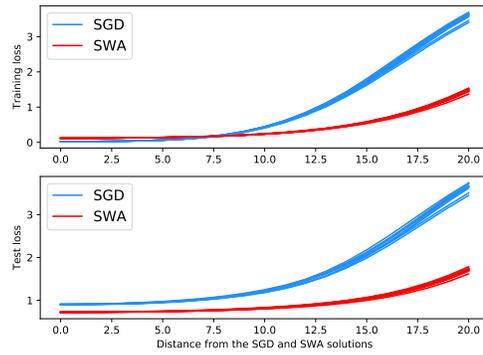


Figure 31. Random ray of SGD and SWA solution

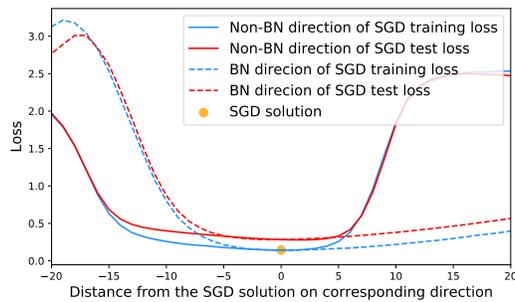


Figure 32. BN and Non-BN directions comparison of ResNet-164 on CIFAR-10

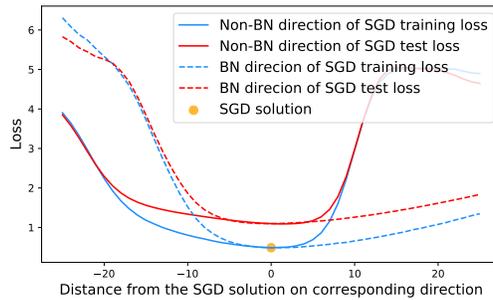


Figure 33. BN and Non-BN directions comparison of ResNet-110 on CIFAR-100

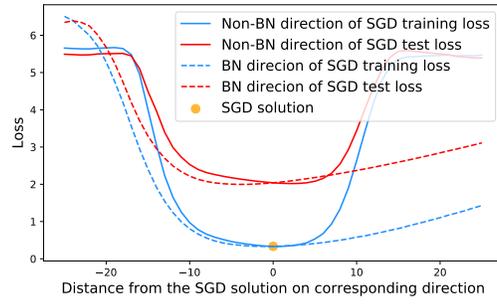


Figure 34. BN and Non-BN directions comparison of DenseNet-100 on CIFAR-100