The Regularization Effects of Anisotropic Noise in Stochastic Gradient Descent

Zhanxing Zhu [*1,2] **Jingfeng Wu** [*3] **Bing Yu** [3] **Lei Wu** [3] **Jinwen Ma** [3]

Abstract

Understanding the generalization of deep learning has raised lots of concerns recently, where the learning algorithms play an important role in generalization performance, such as stochastic gradient descent (SGD). Along this line, we particularly study the anisotropic noise introduced by SGD, and investigate its importance for the generalization in deep neural networks. Through a thorough empirical analysis, it is shown that the anisotropic diffusion of SGD tends to follow the curvature information of the loss landscape, and thus is beneficial for escaping from sharp and poor minima effectively, towards more stable and flat minima. We verify our understanding through comparing this anisotropic diffusion with full gradient descent plus isotropic diffusion (i.e. Langevin dynamics) and other types of position-dependent noise.

1. Introduction

As a successful learning algorithm, stochastic gradient descent (SGD) was originally adopted for dealing with the computational bottleneck of training neural networks with large-scale datasets (Bottou, 1991). Its empirical efficiency and effectiveness have attracted lots of attention. And thus, SGD and its variants have become standard workhorse for learning deep models. Besides the aspect of computational efficiency, recently, researchers have started to realize that the noise introduced by SGD impacts the generalization in deep learning, thanks to the enormous researches on the phenomenon that training with a large batch could causes a significant drop of test accuracy (Keskar et al., 2017). Particularly, several works attempted to investigate how the magnitude of the noise influences the generalization during the process of SGD optimization, including the batch size and learning rate (Hoffer et al., 2017; Goyal et al., 2017; Chaudhari & Soatto, 2017; Jastrzębski et al., 2017).

With the assumption of smooth loss function, the generalization bounds were provided based on stability analysis in non-convex settings (Hardt et al., 2016; Kuzborskij & Lampert, 2017). In (Brutzkus et al., 2017), the authors showed that SGD provably generalizes on linearly separable data for two-layer over-parameterized networks, where the generalization bound is independent of network capacity. However, these theoretical works either imposed strong (or even unrealistic) assumptions or the obtained bounds were too loose.

Another line of research interpreted SGD from a Bayesian perspective. In (Mandt et al., 2017; Chaudhari & Soatto, 2017), SGD was interpreted as performing variational inference, where certain entropic regularization involves to prevent overfitting. (Smith & Le, 2018) tried to provide an understanding based on model evidence. These explanations are compatible with the flat/sharp minima argument (Hochreiter & Schmidhuber, 1997; Keskar et al., 2017), since Bayesian inference tends to targeting the region with large probability mass, corresponding to the flat minima.

More recently, it was shown that the noise introduced by SGD is highly anisotropic in (Chaudhari & Soatto, 2017). Particularly, the authors showed that the trajectories of SGD resemble a close loop due to the non-isotropic property of noise. However, what role this anisotropy property of gradient noise plays in generalization was rarely discussed in literature.

In this work, we find that the noise structure of SGD plays an important role in helping optimizer find solutions generalizing well in the context of deep neural networks. We provide an explanation by connecting the noise structure and the curvature of the landscape. It is found that the optimizer with curvature-aware noise can escaping from sharp minima more efficiently, thus converging to flat minima with a higher probability. Moreover, these flat minima typically generalize well according to various researches (Hochreiter & Schmidhuber, 1997; Keskar et al., 2017; Neyshabur et al., 2017; Wu et al., 2017). We also show that Langevin dynamics with well tuned isotropic noise cannot beat stochastic gradient descent, which further confirms the importance of

[*]Equal contribution [1]Center for Data Science, Peking University, Beijing, China [2]Beijing Institute of Big Data Research (BIBDR), Beijing, China [3]School of Mathematical Sciences, Peking University, Beijing, China. Correspondence to: Lei Wu <leiwu@pku.edu.cn>.

noise structure of SGD.

A large number of experiments are designed systematically to justify our understanding on the behavior of the anisotropic diffusion of SGD. We compare SGD with full gradient descent with different types of diffusion noise, including isotropic and position-dependent/independent noise. All these comparisons demonstrate the effectiveness of anisotropic diffusion for good generalization in training deep networks.

2. Problem Setup and Preliminaries

In general, supervised learning usually involves an optimization process of minimizing an empirical loss over training data,

$$L(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i;\boldsymbol{\theta}), y_i),$$

where $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ denotes the training set with N *i.i.d.* samples, the prediction function f is often parameterized by $\boldsymbol{\theta} \in \mathbb{R}^D$, such as deep neural networks. And $\ell(\cdot,\cdot)$ is the loss function, such as mean squared error and cross entropy, typically corresponding to certain negative log likelihood.

Due to the over parameterization and non-convexity, there exist multiple global minima. In practical training of deep neural networks, the exact global minima are often impossible to obtain. Instead, we train the networks such that $L(\boldsymbol{\theta}) < \epsilon$, with ϵ small enough to achieve the approximate optimality. With slightly abuse of naming convention, we call the solution set $S^* := \{\boldsymbol{\theta}^* \,|\, L(\boldsymbol{\theta}^*) < \epsilon, \text{ with } \epsilon \text{ small enough}\}$ as the global minima, which are the subjects we would study in deep learning community. The set S^* consists of a large number of solutions with diverse generalization performance. We call those solutions generalizing well good solutions or minima, vice versa.

In a typical setting of SGD for optimizing $L(\boldsymbol{\theta})$, during each iteration t, a minibatch of training samples with size m are randomly selected, $B_t \subset \{1, 2, \ldots, N\}$. A stochastic gradient is evaluated based on the chosen minibatch, $\tilde{\boldsymbol{g}}(\boldsymbol{\theta}_t) = \sum_{i\in B_t} \nabla_{\boldsymbol{\theta}} \ell(f(\boldsymbol{x}_i;\boldsymbol{\theta}_t), y_i)/m$, which is an unbiased estimator of the full gradient, $\boldsymbol{g}_0(\boldsymbol{\theta}_t) = \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_t)$. Then, the network parameters are updated as follows with some learning rate η_t,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \tilde{\boldsymbol{g}}(\boldsymbol{\theta}_t). \tag{1}$$

Denote $\boldsymbol{g}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \ell((f(\boldsymbol{x};\boldsymbol{\theta}), y)$ for a single data point $(\boldsymbol{x}, y)$, and assume that the size of minibatch is large enough for the central limit theorem to hold, and thus $\tilde{\boldsymbol{g}}(\boldsymbol{\theta}_t)$ follows a Gaussian distribution,

$$\tilde{\boldsymbol{g}}(\boldsymbol{\theta}_t) \sim \mathcal{N}\left(\boldsymbol{g}_0(\boldsymbol{\theta}_t), \frac{1}{m}\hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}_t)\right), \tag{2}$$

and the covariance matrix of the noisy gradient,

$$\hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}_t) \approx \frac{1}{N}\sum_{i=1}^{N} (g(\boldsymbol{\theta}_t;\boldsymbol{x}_i) - \boldsymbol{g}_0(\boldsymbol{\theta}_t))(g(\boldsymbol{\theta}_t;\boldsymbol{x}_i) - \boldsymbol{g}_0(\boldsymbol{\theta}_t))^T. \tag{3}$$

Note that this covariance matrix depends on the model architecture, dataset and the current parameter $\boldsymbol{\theta}_t$. Now we can rewrite the update of SGD as,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \boldsymbol{g}_0(\boldsymbol{\theta}_t) + \frac{\eta_t}{\sqrt{m}}\boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}\left(\mathbf{0}, \hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}_t)\right). \tag{4}$$

For constant learning rate $\eta_t = \eta$, the above iteration can be treated as the numerical discretization of the following stochastic differential equation (Li et al., 2015; Jastrzębski et al., 2017; Chaudhari & Soatto, 2017),

$$d\boldsymbol{\theta}_t = -\nabla L(\theta_t)dt + \sqrt{\frac{\eta}{m}\hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}_t)}dW_t, \tag{5}$$

where $\gamma = \eta/m$ controls the magnitude of noise introduced by mini-batch approximation.

Hoffer et al. (2017) and Jastrzębski et al. (2017) studied the influence of noise magnitude on generalization . However we focus on studying how the anisotropic structure of $\hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta})$ impacts generalization by building the connection between the gradient covariance and the curvature of loss surface, particularly in the context of deep neural networks.

3. Covariance Structure of SGD Noise

In this section, we mainly investigate the structure of gradient covariance, and explore its connection with the curvature of loss surface $\hat{\boldsymbol{H}}(\boldsymbol{\theta}) := \nabla^2 L(\boldsymbol{\theta})$, which is strongly correlated with the generalization ability , as analyzed by Wu et al. (2017) and Jastrzębski et al. (2017).

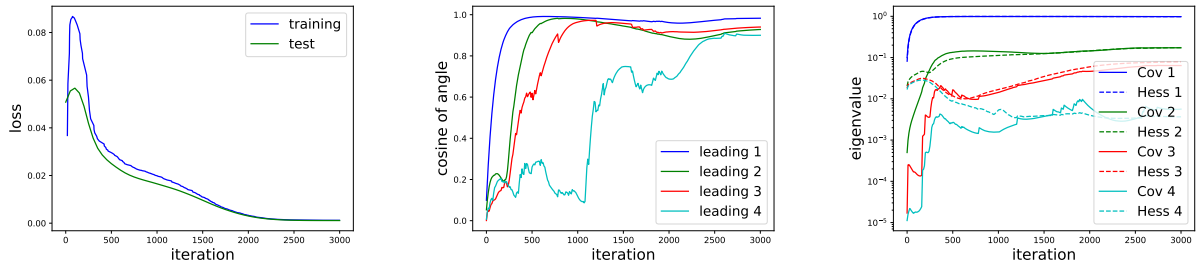
According to the classic statistical theory (Pawitan, 2001, Chap. 8), when evaluating at the true parameter $\boldsymbol{\theta}^*$, we have the exact equivalence between the expected Hessian of negative log likelihood and *Fisher information matrix*, $\boldsymbol{\Sigma}(\boldsymbol{\theta}^*)$,

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}^*) := \mathbb{E}_{(\boldsymbol{x},y)}[\boldsymbol{g}(\boldsymbol{\theta}^*)\boldsymbol{g}(\boldsymbol{\theta}^*)^T] = \mathbb{E}_{(\boldsymbol{x},y)}[\nabla^2_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}^*)], \tag{6}$$

In practice, with the assumptions that the sample size N is large enough (i.e. indicating asymptotic behavior) and the current parameter $\boldsymbol{\theta}_t$ is not far from the ground truth, we can obtain the following approximate equality,

$$\hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}_t) \approx \hat{\boldsymbol{H}}(\boldsymbol{\theta}_t). \tag{7}$$

Clearly, the closeness between the two matrices depends on the sample size N, model family and $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|$. With slightly abuse of notation, we denote the empirical covariance and curvature by $\boldsymbol{H}$ and $\boldsymbol{\Sigma}$ for simplicity.



(a) (b) (c)

Figure 1. **(a)** Training and test loss versus number of iterations **(b)** Cosine similarity of first 4 leading vectors **(c)** first 4 leading eigenvalues of the normalized Hessian and covariance matrix of noisy gradient. The task is to fit an order-3 polynomial with 20 training points. The network architecture is fully connected, 3 hidden layers with total 251 parameters. The network is trained by SGD with mini batch size 5 and learning rate 0.05.

Particularly, in the scenario of modern neural networks, most of the deep models are over-parameterized, i.e. $N \ll D$. Both of $\boldsymbol{H}$ and $\boldsymbol{\Sigma}$ are highly ill-conditioned and anisotropic, as shown by Sagun et al. (2017) and Chaudhari & Soatto (2017), respectively, i.e. only a small portion of leading eigenvalues of the two matrices are significant, and most of eigenspectrum concentrates around zero. Taking these properties of neural network models into consideration, whether the exact equality (7) holds or not is not clear yet. Moreover, for such over-parameterized and nonlinear systems, there exists no analytic tools to characterize the distance between the two matrices rigidly. Therefore, we resort to empirical investigation over the relationship between the two key matrices in various deep models under consideration. Still, this is the starting point for us to analyze the regularization effects of the anisotropic noise introduced by SGD, since it might closely connect with the curvature of the loss surface.

Measuring the Geometric Similarity of Two Matrices. We now empirically explore the approximate equivalence (7) during the SGD training of neural networks. Due to the high ill-conditioning and anisotropy of the gradient covariance and Hessian, the main geometric properties of the two matrices can be captured by their leading eigen directions.

To this end, we firstly implement eigen decomposition over $\boldsymbol{H}$ and $\boldsymbol{\Sigma}$,

$$\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T, \quad \boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Gamma}\boldsymbol{V}^T, \tag{8}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_D], \boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_D]$, each column vector of $\boldsymbol{U}$ and $\boldsymbol{V}$ denoting an eigen vector of the corresponding matrix. The diagonal matrices $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_D)$ and $\boldsymbol{\Gamma} = \text{diag}(\gamma_1, \gamma_2, \ldots, \gamma_D)$ consists of D eigenvalues in a non-increasing order.

To measure the geometric similarity between the two matrices, we compute the cosine distance between the leading eigen vectors of the two matrices,

$$\cos(\alpha) = \frac{\boldsymbol{u}_i^T \boldsymbol{v}_i}{\|\boldsymbol{u}_i\|_2 \|\boldsymbol{v}_i\|_2}, \tag{9}$$

where α is the angle between the two eigenvectors, $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$.

Figure 1 presents the cosine similarity evolving with the number of iterations during SGD training for a regression task. It can be easily observed that during the later stage of the training process, i.e. near the global minima, the cosine similarity between leading eigen vectors of $\boldsymbol{H}$ and $\boldsymbol{\Sigma}$ approaches 1. It demonstrates consistency with the approximate equality between $\boldsymbol{H}$ and $\boldsymbol{\Sigma}$ in Eq (7). Though the magnitude of the leading eigenvalues of the two matrices have different scales, after normalization by their individual Frobenous norm, we find the normalized eigenvalues are extremely similar. This again confirms the similar geometric properties between gradient covariance and Hessian. Note that these findings are shared among the fully-connected networks with different number of hidden layers and hidden units for one-dimensional regression tasks.

4. The Impacts of Anisotropic Diffusion on Generalization

4.1. Significance of Noise in SGD Dynamics

We start analyzing the impacts of anisotropic diffusion on generalization by investigating the magnitude of the noise introduced by SGD. This is necessary since that in the late stage of optimization, the gradient approaches zero, but the stochastic gradient noise might still exist, which would influence the behavior of the optimization dynamics significantly.

To study the relative difference between gradient mean and variance, we measure the noise magnitude by the expected L_2 norm of $\boldsymbol{\epsilon}_t$ in Eq. (4). Since we can write $\boldsymbol{\epsilon}_t = \boldsymbol{V}\sqrt{\boldsymbol{\Gamma}}\boldsymbol{z}$,

with $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ being a standard Gaussian noise, we have

$$\boldsymbol{\epsilon}_t^T \boldsymbol{\epsilon}_t = \left(\boldsymbol{V}\sqrt{\boldsymbol{\Gamma}}\boldsymbol{z}\right)^T \left(\boldsymbol{V}\sqrt{\boldsymbol{\Gamma}}\boldsymbol{z}\right) = \boldsymbol{z}^T \boldsymbol{\Gamma} \boldsymbol{z} = \sum_{d=1}^{D} \gamma_d z_i^2. \tag{10}$$

Then the expected squared norm of $\boldsymbol{\epsilon}_t$ can be evaluated as

$$\mathbb{E}[\boldsymbol{\epsilon}_t^T \boldsymbol{\epsilon}_t] = \sum_{d=1}^{D} \gamma_d \mathbb{E}[z_i^2] = \sum_{d=1}^{D} \gamma_d = \mathrm{Tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t)). \tag{11}$$

Here, for a regression task (the same setting as Figure 1), we compare the norm of gradient mean $\boldsymbol{g}_0(\boldsymbol{\theta}_t)$ and the significance of gradient noise measured by $\sqrt{\eta_t \mathbb{E}[\boldsymbol{\epsilon}_t^T \boldsymbol{\epsilon}_t]/m}$ during the optimization, and Figure 2 displays the result.

It can be easily observed that *the noise variance dominates over the mean of the gradient overwhelmingly, especially in the late iterations*. This observation is consistent with the findings in (Shwartz-Ziv & Tishby, 2017). For more experimental results on other architectures and datasets, see the Supplementary Materials.

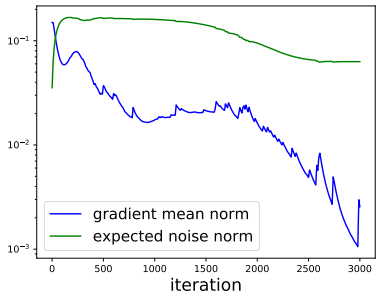


Figure 2. L_2 norm of gradient mean, $\|\boldsymbol{g}_0(\boldsymbol{\theta}_t)\|$, and significance of noise $\sqrt{\eta_t \mathbb{E}[\boldsymbol{\epsilon}_t^T \boldsymbol{\epsilon}_t]/m}$ during the training using SGD for the same regression task as Figure 1.

4.2. The Effects of Stochastic Gradient Noise on Generalization

Given the significance of noise analyzed in Section 4.1, the diffusion introduced by noise probably takes a non-negligible effect on the dynamics of SGD, particularly in later iterations. More importantly, taking both of SGD dynamics in Eq. (4) and the covariance structure in Eq. (7) into consideration, we can easily observe that, *the diffusion of SGD approximately follows the curvature information of the loss surface, especially near the minima*.

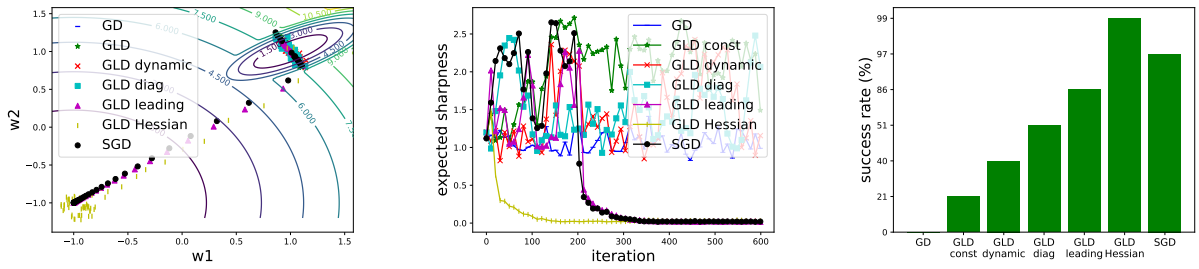
Thanks to this nice property, SGD adaptively assigns a large portion of noise variance along the directions of top eigenvectors of the Hessian matrix, i.e. the sharpest directions. When trapped in certain sharp/poor minima, SGD can contribute all the kinetic energy introduced by noise along the sharp directions due to the highly anisotropic curvature, rather than spread along those already flat directions. Expectedly, this characteristic can drive SGD to escape from sharp minima more efficiently, thus converging to a more stable/flat minima, which typically generalize better.

Now we summarize several remarkable characteristics of SGD's anisotropic diffusion,

- SGD is originally utilized for overcoming the computational bottleneck in training large neural networks. Surprisingly, SGD also brings additional benefits over good generalization for free, since the gradient noise naturally adapts with the curvature and favors the flat minima that help to generalize well.
- Compared with gradient descent with isotropic Gaussian diffusion, such as standard Langevin dynamics, the curvature-aware anisotropic noise steers the SGD to diffuse much more aggressively along the sharp directions. Since $\nabla^2 L(\boldsymbol{\theta})$ is highly ill-conditioned, the isotropic diffusion inevitably wastes the most of energy along the flat direction, which helps little for the optimizer find the flat minima.

In order to justify the above arguments and demonstrate the effectiveness of the anisotropic diffusion of SGD, we compare SGD with a variety of optimization dynamics,

1. Standard gradient descent (**GD**), $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \boldsymbol{g}_0(\boldsymbol{\theta}_t)$;
2. Gradient descent with isotropic diffusion (standard Langevin dynamics, abbreviated as **GLD const**), $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \boldsymbol{g}_0(\boldsymbol{\theta}_t) + \eta_t \boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ and σ^2 is a constant to be tuned.
3. Gradient descent with dynamic and isotropic diffusion (denoted as **GLD dynamic**), $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \boldsymbol{g}_0(\boldsymbol{\theta}_t) + \frac{\eta_t}{\sqrt{m}} \boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2(\boldsymbol{\theta}_t)\boldsymbol{I})$ and $\sigma^2(\boldsymbol{\theta}_t) = \mathrm{Tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t))/D$; This type of isotropic diffusion can guarantee the expected norm of noise vector to equals to that of SGD in each iteration. GLD dynamic can be interpreted as a type of adaptive simulated annealing scheme, since the noise magnitude is dynamically adjusted.
4. Gradient descent with diagonal diffusion (denoted as **GLD diagonal**), $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \boldsymbol{g}_0(\boldsymbol{\theta}_t) + \frac{\eta_t}{\sqrt{m}} \boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(\sigma_i^2(\boldsymbol{\theta}_t)))$ and the diagonal variance $\sigma_i^2(\boldsymbol{\theta}_t) = \Sigma_{ii}(\boldsymbol{\theta}_t)$, $i = 1, 2, \ldots, D$. This diffusion ignores the correlations among different freedom, which should be a bad approximation of $\boldsymbol{\Sigma}$ due to the low-rank property of $\boldsymbol{\Sigma}$.
5. Gradient descent with diffusion noise of leading eigen directions of $\boldsymbol{\Sigma}(\boldsymbol{\theta}_t)$ (**GLD leading**), $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t -$



(a) (b) c

Figure 3. **(a)** The trajectory of each compared dynamics for escaping from sharp minima in 2-dimensional case, the noise std of GLD const is $\sigma = 2$, which is already greater than the noise std of GLD dynamic. **(b)** Expected sharpness versus number of iterations, where the noise variance for parameter perturbation $s = 0.1$. **(c)** Success rate of arriving the flat solution in 600 iterations. The learning rate is same for all the compared methods, $\eta_t = 0.1$, and batch size $m = 1$. Since the curvature of the loss function is not highly ill-conditioned, we let $k = 2$ for GLD leading method in Eq.(12), i.e. the full covariance matrix is used. For GLD Hessian, we set $\sigma_H^2 = 1$ for simiplicty.

$\eta_t \boldsymbol{g}_0(\boldsymbol{\theta}_t) + \frac{\eta_t}{\sqrt{m}}\boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \tilde{\boldsymbol{\Sigma}}_k(\boldsymbol{\theta}_t))$, and

$$\tilde{\boldsymbol{\Sigma}}_k(\boldsymbol{\theta}_t) = \sum_{l=1}^{k} \gamma_k \boldsymbol{v}_k \boldsymbol{v}_k^T \tag{12}$$

is a low-rank approximation of $\boldsymbol{\Sigma}(\boldsymbol{\theta}_t)$. In the context of deep learning, a small $k \ll D$ should be able to make $\tilde{\boldsymbol{\Sigma}}_k(\boldsymbol{\theta}_t)$ approximate $\boldsymbol{\Sigma}(\boldsymbol{\theta}_t)$ well due to the high anisotropy and ill-conditioning.

We hypothesis that, similar with SGD, the gradient descent diffusion following the leading eigen directions should own the property of fast and effective escaping from sharp/poor minima.

6. Gradient descent with Hessian as diffusion noise (**GLD Hessian**), $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \boldsymbol{g}_0(\boldsymbol{\theta}_t) + \eta_t \boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \sim \mathcal{N}\left(\mathbf{0}, \sigma_H^2 \boldsymbol{H}_+(\boldsymbol{\theta}_t)\right)$. The noise magnitude σ_H^2 can be tuned, and the covariance matrix $\boldsymbol{H}_+ := \sum_{i=1}^{D} |\lambda_i| \boldsymbol{u}_i \boldsymbol{u}_i^T$, which reverses the negative eigenvalue of $\boldsymbol{H}$ in the non-convex area.

   Since the diffusion of GLD Hessian exactly follows the loss curvature, it could potentially escape from the minima more effectively without wasting time diffusing along flat directions. Due to the approximate equality between Hessian and gradient covariance, We hypothesis that GLD Hessian should exhibit the similar even superior behavior with SGD for reaching the flat minima, since the noise captures more accurate curvature information than SGD.

Two-dimensional Experiment We design a 2-D toy example $L(w_1, w_2)$ with two basins, a small one and a large one, corresponding to a sharp and flat minima, $(1, 1)$ and $(-1, -1)$, respectively, both of which are global minima. Please refer to Supplementary Material for the details. We initialize all the optimizers with sharp minima $(w_1, w_2) = (1, 1)$, and then run them to minimize the loss for comparing their behaviors.

Figure 3(a) shows the trajectories of all the dynamics for escaping from the sharp minima $(1, 1)$ towards the flat one $(-1, -1)$, and Figure 3(b) depicts the sharpness versus number of iterations. In this paper, we adopt the following sharpness, $\mathbb{E}_{\boldsymbol{\nu}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}\left[L(\boldsymbol{\theta}+\boldsymbol{\nu})\right] - L(\boldsymbol{\theta})$, which is defined in (Neyshabur et al. (2017), Eq.(7)). GLD Hessian as expected converges to the flat minimum fastest, while SGD and GLD leading take the second places. Moreover, the other optimizers even those with isotropic noise appear to escape from sharp minimum much slower. This partially justifies that those curvature-aware diffusions do flee from the sharp basin much more efficiently. Those isotropic and diagonal noises in this case cannot capture the curvature information accurately.

Note that Figure 3 (a) and (b) only presents a single run of the experimental results. To make the previous analysis more reliable, we run all these dynamics for 100 times by limiting the maximum number of iterations as 600. And then collect the success rate of arriving the flat minimum of each optimizer, the result is shown in Figure 3 (c), which again confirms the effectiveness of curvature-aware optimizers, thus explaining why the anisotropic noise can encourages SGD to converge towards flat solutions generalizing better.

In the following section, we conduct more experiments on real-world datasets to further justify that the special structure of SGD noise plays an important role for the regularization effects of SGD in deep neural networks.

5. Empirical Comparison between SGD and Other Optimization Dynamics

In this section, we mainly compare SGD with other types of optimization dynamics equipped with different diffusion mechanisms which are described in Section 4.2. In the first part, we conduct a full comparison for small neural networks and noisy data, where the covariance matrix $\boldsymbol{\Sigma}$ can be stored

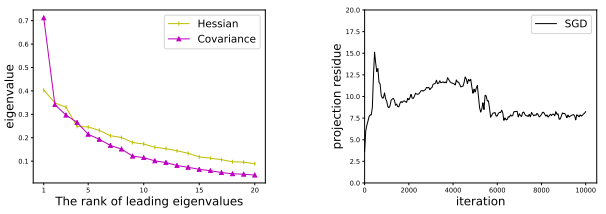
and tractably spectrally decomposed. In the second part, learning VGG11 over CIFAR-10 is considered to show that anisotropic noise is superior to isotropic noise in terms of generalization performance.

5.1. Small Networks with Noisy Data

To better demonstrate the generalization difference among optimization dynamics, we construct a noisy training set based on FashionMNIST dataset[1]. Concretely, the training set consist of 1000 images with correct labels, and another 200 images with randomly wrong labels. All the test set are with clean labels. A small LeNet-like networks is utilized such that the eigen decomposition over gradient covariance matrix computational is feasible. This net consists of two convolutional layers and two fully-connected layers . The total number of parameters is $11,330$.

In the following, we first investigate the geometric similarity between Hessian and the covariance matrix of stochastic gradient noise. Then we compare the behaviors of different optimization dynamics to verify the benefits of anisotropic diffusion noise for generalization.

The similarity between gradient covariance and Hessian. Previously in Section 3, we quantify the similarity between the two matrices by measuring the angle between the two eigenvectors $(\boldsymbol{u}_i, \boldsymbol{v}_i)$, according to the exact decreasing order of the eigenvalue. However, when we check eigenspectrum of the normalized $\boldsymbol{H}(\boldsymbol{\theta}_t)$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta}_t)$, we find the eigen gaps between the leading eigenvalues is not large enough, as shown in Figure 4(a), where $k = 20$ leading eigenvalues are presented. Only comparing the eigenvectors pair $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ one by one might ignore the misalignment between the eigenvectors due to the not well separated eigen gaps.



(a) (b)

Figure 4. **(a)** FashionMNIST data: 20 leading eigenvalues of gradient covariance and Hessian. **(b)** The square of Frobenius norm of the "projection residual".

To handle the potential misalignment, we compare the similarity between the orthogonal subspaces spanned by the k leading eigenvectors, $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$, respectively. This similarity can be quantified via the squared Frobenius norm of the "projection residual", defined as

$$e(\boldsymbol{U}_k, \boldsymbol{V}_k) = \|\boldsymbol{U}_k - \boldsymbol{V}_k \boldsymbol{V}_k^T \boldsymbol{U}_k\|_F^2. \tag{13}$$

When the two orthogonal subspaces are overlapped with each other, $e = 0$; when orthogonal with each other, $e = k$, and therefore $e \in [0, k]$. We use this quantity $e(\boldsymbol{U}_k, \boldsymbol{V}_k)$ to measure the distance between the gradient covariance and loss curvature near the minima. We run SGD and plot the squared norm of the "projection residual" versus number of iterations, see Figure 4(b), when first $k = 20$ leading eigenvectors are considered. As observed from the plot, the projection residual becomes small and stable in the later stage of optimization, indicating the geometric similarity between $\boldsymbol{H}$ and $\boldsymbol{\Sigma}$.

Benefits of anisotropic noise. Provided that the similarity between the $\boldsymbol{H}$ and $\boldsymbol{\Sigma}$, we now show that the curvature-following noise can indeed help the optimizer to get away from poor and sharp minima towards flat minima that generalize well.

We firstly run the standard gradient decent for 3000 iterations to arrive at the parameters $\boldsymbol{\theta}^*_{GD}$ near the global minima with near zero training loss and 100% training accuracy, and then all other compared methods are initialized with $\boldsymbol{\theta}^*_{GD}$ and run for optimization with the same learning rate $\eta_t = 0.07$ for fair comparison[2].

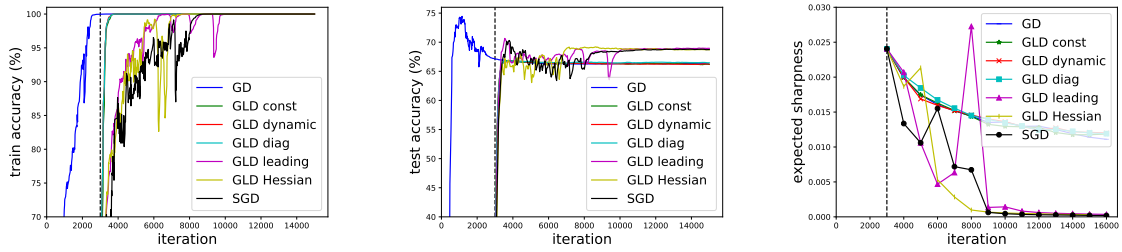
The settings for each compared approaches are as follows. The noise variance for GLD const has already been tuned as optimal by grid search. For GLD leading, we set $k = 20$ for comprising the computational cost and approximation accuracy. As for GLD Hessian, to reduce the expensive evaluation of such a huge Hessian in each iteration, we update the Hessian every 10 iterations; moreover, restricted by computational resource for tuning the noise magnitude σ_H^2, we simply set $\sigma_H^2 = \text{Tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t))/(m\text{Tr}(\boldsymbol{H}(\boldsymbol{\theta}_t))$ to guarantee that it has the same expected noise norm as that of SGD.

The entire dynamic processes are plotted in Figure 5. The following key observations and insights can be achieved,

- Given such an over-parameterized CNN, all the optimization dynamics can easily reach 100% training accuracy and near zero cross entropy loss. However, the solutions that they find have different flatness and generalization performance.
- Gradient descent, GLD const, GLD dynamics, and GLD diagonal have difficulty in moving away from the sharp minima that GD has found. The first three

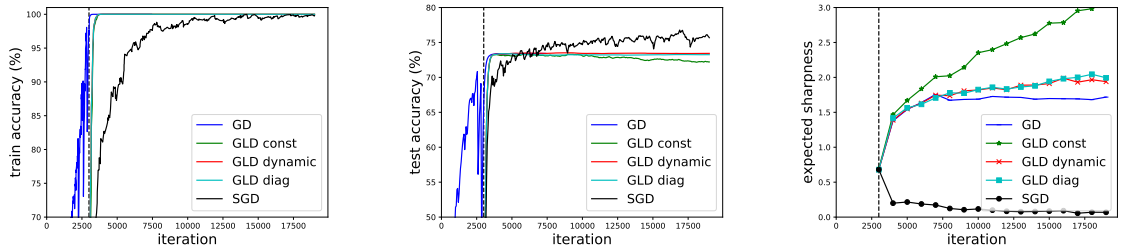
[1] https://github.com/zalandoresearch/fashion-mnist

[2] In fact, in our experiment, we test the equally spacing learning rates in the range $[0.01, 0.1]$, and the final results are consistent with each other.



(a) (b) (c)

Figure 5. **(a)** FashionMNIST data: the test accuracy of each compared dynamics with the initialization at the global minima $\boldsymbol{\theta}^*_{GD}$ found by GD, starting from iteration 3000 marked with a vertical dashed line. The noise std of GLD const has been tuned as optimal, $\sigma = 0.001$. **(b)** Expected sharpness versus number of iterations, where the noise std for parameter perturbation $s = 0.01$. The learning rate is same for all the compared methods, $\eta_t = 0.07$, and batch size $m = 20$.



(a) (b) (c)

Figure 6. **(a)** CIFAR-10 data: the test accuracy of each compared dynamics with the initialization at the global minima $\boldsymbol{\theta}^*_{GD}$ found by GD, starting from iteration 3000 marked with a vertical dashed line. The noise std of GLD const has been tuned as optimal, $\sigma = 0.001$. **(b)** Expected sharpness versus number of iterations, where the noise std for parameter perturbation $s = 0.01$. The learning rate is same for all the compared methods, $\eta_t = 0.1$, and batch size $m = 100$.

methods equip with no diffusion or only isotropic diffusion mechanisms, and GLD diagonal does not capture the couplings between different dimensions, due to the low-rank properties of $\boldsymbol{H}$ and $\boldsymbol{V}$. All these methods fails to fully consider the curvature information during the diffusion, leading to inefficient escaping.

- Favorably, SGD , GLD leading and GLD Hessian all converge to flat minima that generalize better. Moreover GLD Hessian escapes from the sharp minimum slightly better than SGD and GLD leading. It is worth mentioning that the workability of GLD leading stems from the anisotropic property of both stochastic gradient noise and the curvature of landscape.

- By comparison of these two categories of learning methods, we can conclude that the curvature-adapted diffusion does help to find flat minima more efficiently and yields better generalization performance. In general, treating the noise of SGD as isotropic one cannot explain the implicit regularization property of SGD, due to the highly non-isotropic nature of $\boldsymbol{\Sigma}$ and $\boldsymbol{H}$.

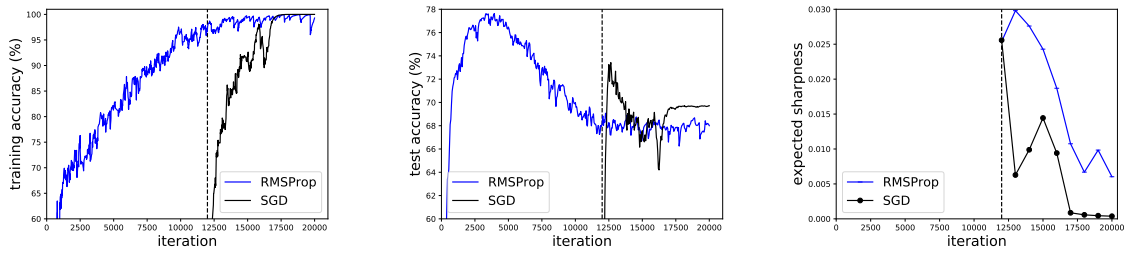
5.2. Large Networks with Clean Data

In this part, we test standard VGG11 network on CIFAR-10 dataset. Since the number of total parameters is around ten millions, the estimated gradient covariance matrix and Hessian cannot even be stored and implemented spectral decomposition. We only compare SGD with other GD, GLD const, GLD dynamic and GLD diag. In order to make a fair comparison and exclude other factors that might affect generalization, we does not add any architecture modification or extra training strategies, such as batch normalization and Dropout. Same as the settings in fashionMNIST experiment, we run all the compared optimization dynamics initialized with a near the end point of gradient descent. And the results are displayed in Figure 6.

As we can see, SGD converges to a minimum that is more flat and generalizable than all the other optimizers. Moreover, the optimizers with isotropic and diagonal noise do not provide us solutions better than full batch gradient descent. These observations further confirm our previous argument in this realistic scenario.

6. Discussions

Benefits of considering covariance structure. As mentioned in Section 1, previous works on SGD for deep learning typically assumed the norm of the covariance matrix is upper bounded, such that the dynamics of SGD can be analyzed easily from its continuous form, a stochastic differential equation. However, this assumption ignores the



(a) (b) (c)

Figure 7. Training, test accuracy and expected sharpness, shown in (a), (b) and (c), respectively. SGD is initialized with the solution achieved by RMSProp at $12,000$th iteration on FashionMNIST data, marked with the vertical dashed line.

covariance structure, as we have shown in this work, which has significant effects on its dynamics behaviors and generalization performance as well.

The key observation on connecting gradient noise structure with curvature of the loss landscape, especially near the minima shown in Eq. (7), provides a new perspective on understanding the reason that why SGD can achieve good generalization in practice. Our work is an initial attempt to reveal the non-negligible benefits of SGD's covariance structure. More theoretical explorations are needed along this direction.

Effects of learning rate and batch size. As seen from the SGD dynamics in Eq. (4), when the learning rate is too small or batch size is overly large, the magnitude of gradient noise will become small, and thus effects of covariance structure is not obvious as before. In these cases, SGD often needs long time for diffusion towards flat minina to obtain better solutions, as shown in existing research (Keskar et al., 2017; Hoffer et al., 2017; Jastrzębski et al., 2017).

Understanding the generalization performance of adaptive variants of SGD. Wilson et al. (2017) recently investigated the adaptive variants of SGD with certain diagonal preconditioning, such as AdaGrad, RMSProp and Adam. It showed that empirically all these adaptive methods exhibit a worse generalization capability than SGD, though they enjoy fast convergence speed during the early stage of training for deep networks. According to the findings in our work, the difference between the gradient noise covariance matrices of the adaptive methods and non-adaptive ones, such as SGD, might provide a plausible explanation on their generalization difference.

Take RMSProp as an example, its learning dynamics is as follows,

$$\boldsymbol{r}_{t+1} = \rho \boldsymbol{r}_t + (1-\rho)\tilde{\boldsymbol{g}}(\boldsymbol{\theta}_t) \circ \tilde{\boldsymbol{g}}(\boldsymbol{\theta}_t) \tag{14}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta_t}{\sqrt{\boldsymbol{r}_t + \delta}} \tilde{\boldsymbol{g}}(\boldsymbol{\theta}_t), \tag{15}$$

where "$\circ$" denotes the element-wise product, the hyperparameter $\rho \in (0, 1)$ and δ is a small constant to avoid the numerical issue. We can rewrite the second update step as,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta_t}{\sqrt{\boldsymbol{r}_t + \delta}} \boldsymbol{g}_0(\boldsymbol{\theta}_t) + \frac{\eta_t}{\sqrt{m}} \boldsymbol{\epsilon}_t, \tag{16}$$

where the diffusion noise,

$$\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(\frac{1}{\sqrt{\boldsymbol{r}_t + \delta}}) \boldsymbol{\Sigma}(\boldsymbol{\theta}_t)). \tag{17}$$

Clearly, the diagonal preconditioning introduced in RMSProp modified the covariance structure dramatically. Its diffusion dynamics does not follow the curvature of loss surface any more, and may be less effective on escaping from sharp minima than SGD. This might partially explain its worse generalization performance in practice. The argument can also be applied to other variants of SGD with adaptive preconditioning.

To see the different diffusion performance between SGD and RMSProp, for FashionMNIST dataset (the same experimental settings as Section 5.1), we firstly run RMSProp to obtain a solution near the global minima. Then initialized with the RMSProp solution, SGD is run to check whether SGD can escape the poor minima towards a better one. We plot the training and test accuracy in Figure 7.

Thus, it is suggested that, in the early stage of training, adaptive methods can be used for fast training, and the optimizer should be switched to SGD in later iterations such that parameters are easy to diffuse towards flatter minima for better generalization.

Insights on the loss landscape. Through our through analysis, the effectiveness of SGD also reveals some important properties of the loss landscape. The SGD's behavior of fast escaping from sharp minima shows that, along the sharp directions the barrier between the current and neighborhood minima is not that significantly high, otherwise, SGD might also fail to escape. This good property of the loss landscapes shared by most of popular architectures and datasets guarantee the success of the optimization algorithms.

7. Conclusion

In this work, extensive experiments are designed and conducted to investigate how the noise structure of SGD affects the solutions founded in terms of generalization. We first show that the covariance matrix of SGD noise approximately equal to Hessian matrix, especially during late stage of optimization. Based on this observation, we partially attribute the effectiveness of SGD to its Hessian-like noise, which drives SGD to diffuse more aggressively along sharp directions. Therefore, SGD escapes from sharp minima faster than GD, thus converging to a flat solution generalizing better. We also justify this understanding by enormous experiments on both toy and real scenarios.

Our study also shows that isotropic noise helps little for escaping from sharp minima, due to the highly anisotropic nature of landscape. This indicates that it is not sufficient to analyze SGD by treating it as an isotropic diffusion over landscape (Zhang et al., 2017; Mou et al., 2017). A better understanding of this out-of-equilibrium behavior (Chaudhari & Soatto, 2017) is on demand.

References

Bottou, Léon. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8), 1991.

Brutzkus, Alon, Globerson, Amir, Malach, Eran, and Shalev-Shwartz, Shai. Sgd learns over-parameterized networks that provably generalize on linearly separable data. *arXiv preprint arXiv:1710.10174*, 2017.

Chaudhari, Pratik and Soatto, Stefano. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. *arXiv preprint arXiv:1710.11029*, 2017.

Goyal, Priya, Dollár, Piotr, Girshick, Ross, Noordhuis, Pieter, Wesolowski, Lukasz, Kyrola, Aapo, Tulloch, Andrew, Jia, Yangqing, and He, Kaiming. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1225–1234, 2016.

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

Hoffer, Elad, Hubara, Itay, and Soudry, Daniel. 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. 2017.

Jastrzębski, Stanisław, Kenton, Zachary, Arpit, Devansh, Ballas, Nicolas, Fischer, Asja, Bengio, Yoshua, and Storkey, Amos. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 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 *In International Conference on Learning Representations (ICLR)*, 2017.

Kuzborskij, Ilja and Lampert, Christoph. Data-dependent stability of stochastic gradient descent. *arXiv preprint arXiv:1703.01678*, 2017.

Li, Qianxiao, Tai, Cheng, et al. Stochastic modified equations and adaptive stochastic gradient algorithms. *arXiv preprint arXiv:1511.06251*, 2015.

Mandt, Stephan, Hoffman, Matthew D, and Blei, David M. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

Mou, Wenlong, Wang, Liwei, Zhai, Xiyu, and Zheng, Kai. Generalization bounds of sgld for non-convex learning: Two theoretical viewpoints. *arXiv preprint arXiv:1707.05947*, 2017.

Neyshabur, Behnam, Bhojanapalli, Srinadh, Mcallester, David, and Srebro, Nati. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems 30*, pp. 5949–5958. 2017.

Pawitan, Yudi. *In all likelihood: statistical modelling and inference using likelihood*. Oxford University Press, 2001.

Sagun, Levent, Evci, Utku, Guney, V Ugur, Dauphin, Yann, and Bottou, Leon. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Shwartz-Ziv, Ravid and Tishby, Naftali. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Smith, Samuel L. and Le, Quoc V. A bayesian perspective on generalization and stochastic gradient descent. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJij4yg0Z.

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 30*, pp. 4151–4161. 2017.

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

Zhang, Yuchen, Liang, Percy, and Charikar, Moses. A hitting time analysis of stochastic gradient langevin dynamics. *arXiv preprint arXiv:1702.05575*, 2017.

Appendices

The following appendix contains the experiment details and some additional results. To avoid over-tuning of one method over the others, all learning rates in each experiments are kept constant during optimization.

A. Regression Task

Dataset

- Training Set: 20 data points sampled equidifferently from order-3 polynormial $\{(x, y) : y = (x+0.5)x(x-0.5), x \in [-1, 1]\}$.
- Test Set: 100 data points sampled equidifferently from order-3 polynormial $\{(x, y) : y = (x + 0.5)x(x - 0.5), x \in [-1, 1]\}$.

Models

- *251 parameters (main paper)*: Fully connected network of architecture $1 - 10 - 10 - 10 - 1$, with ReLU activation function, which is trained by SGD, with learning rate $\eta = 0.05$ and batch size $m = 5$.
- *141 parameters*: Fully connected network of architecture $1 - 10 - 10 - 1$, with ReLU activation function. which is trained by SGD, with learning rate $\eta = 0.05$ and batch size $m = 5$.
- *103 parameters*: Fully connected network of architecture $1 - 6 - 6 - 6 - 1$, with ReLU activation function, which is trained by SGD, with learning rate $\eta = 0.1$ and batch size $m = 5$.

These experiments are implemented by TensorFlow 1.5.0.

Additional Results To support our conclusion in main paper, here we perform more experiments on another two network architectures described above. See Figure 8 and Figure 9.

B. Two-dimensional Toy Example

Loss Surface The loss surface $L(w_1, w_2)$ is constructed as following:

$$
\begin{aligned}
&s_1 = w_1 - 1 - x_1, \\
&s_2 = w_2 - 1 - x_2, \\
&\ell(w_1, w_2; x_1, x_2) = \min\{10(s_1 \cos\theta - s_2 \sin\theta)^2 \\
&+ 100(s_1 \cos\theta + s_2 \sin\theta)^2, (w_1 - x_1 + 1)^2 + (w_2 - x_2 + 1)^2\}, \\
&L(w_1, w_2) = \frac{1}{N}\sum_{k=1}^{N} \ell(w_1, w_2; x_1^k, x_2^k).
\end{aligned}
$$

where

$$
\begin{aligned}
\theta &= \frac{5}{6}\pi, \\
N &= 100, \\
x_i^k &\sim \mathcal{N}(0, 0.01), i = 1, 2.
\end{aligned}
$$

Hyperparameters All learning rates equal to 0.1. The batch size of **SGD** is $m = 1$. The noise std of **GLD constant** is $\sigma = 2.0$, which is greater than the noise std of **GLD dynamic**. The noise term coefficient of **GLD Hessian** is $\sigma_H = 1.0$.

Estimation of Sharpness Sharpness is estimated by

$$
\frac{1}{M}\sum_{j=1}^{M} L(\boldsymbol{\theta} + \boldsymbol{\xi}_j) - L(\boldsymbol{\theta}), \qquad \boldsymbol{\xi}_j \in \mathcal{N}(0, \sigma^2 I),
$$

with $M = 100$ and $\sigma = 0.1$.

These experiments are implemented by PyTorch 0.3.0.

C. Small CNN with Noisy Data

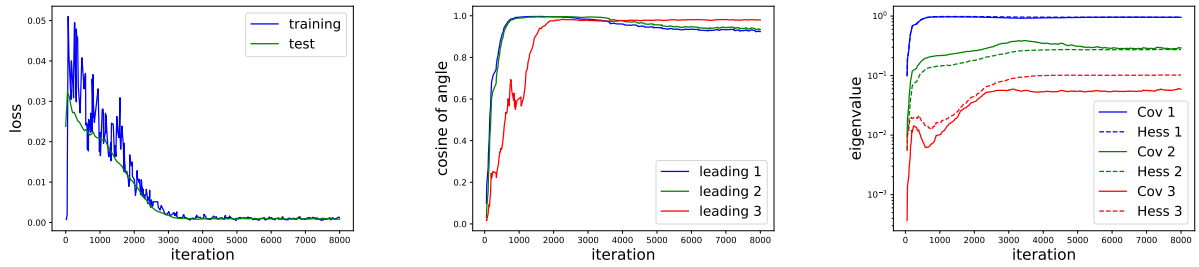
Dataset Our training set consists of 1200 examples randomly sampled from original FashionMNIST training set, and we further specify 200 of them with randomly wrong labels. The test set is same as the FashionMNIST test set.

Model Network architecture:

$$
\begin{aligned}
\text{input} \Rightarrow \text{conv1} \Rightarrow \text{max\_pool} \Rightarrow \text{ReLU} \Rightarrow \text{conv2} \Rightarrow \text{max\_pool} \\
\Rightarrow \text{ReLU} \Rightarrow \text{fc1} \Rightarrow \text{ReLU} \Rightarrow \text{fc2} \Rightarrow \text{output}.
\end{aligned}
$$

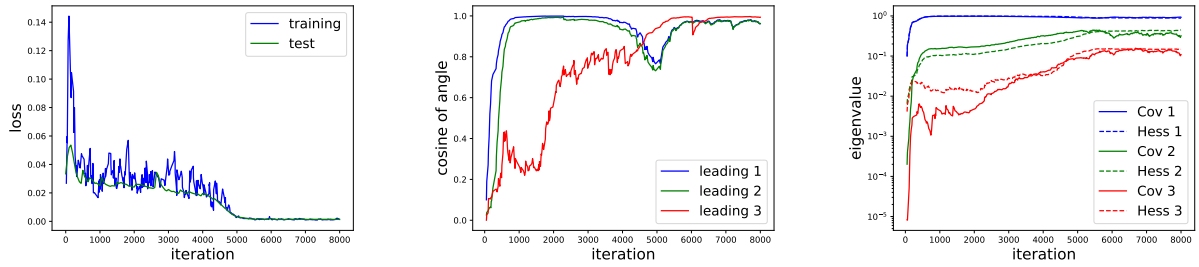
Both two convolutional layers use 5×5 kernels with 10 channels and no padding. The number of hidden units between fully connected layers are 50. The total number of parameters of this network are $11,330$.

Training details



(a) (b) (c)

Figure 8. **(a)** Training and test loss versus number of iterations **(b)** Cosine similarity of first 3 leading vectors **(c)** first 3 leading eigenvalues of the normalized Hessian and covariance matrix of noisy gradient. The network architecture is fully connected, 2 hidden layers with total 141 parameters.



(a) (b) (c)

Figure 9. **(a)** Training and test loss versus number of iterations **(b)** Cosine similarity of first 3 leading vectors **(c)** first 3 leading eigenvalues of the normalized Hessian and covariance matrix of noisy gradient. The network architecture is fully connected, 3 hidden layers with total 103 parameters.

- **GD**: Learning rate $\eta = 0.1$. We tuned the learning rate (in diffusion stage) in a wide range of $\{0.5, 0.2, 0.15, 0.1, 0.09, 0.08, \ldots, 0.01\}$ and no improvement on generalization.
- **GLD constant**: Learning rate $\eta = 0.07$, noise std $\sigma = 10^{-3}$. We tuned the noise std in range of $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ and no improvement on generalization.
- **GLD dynamic**: Learning rate $\eta = 0.07$.
- **GLD diagnoal**: Learning rate $\eta = 0.07$.
- **GLD leading**: Learning rate $\eta = 0.07$, number of leading eigenvalues $k = 20$, mini batch size $m = 20$. We first randomly divide the training set into 60 mini batches containing 20 examples, and then use those minibatches to estimate covariance matrix.
- **GLD Hessian**: Learning rate $\eta = 0.07$, number of leading eigenvalues $= 20$, update frequence $f = 10$. Do to the limit of computational resources, we only update Hessian matrix every 10 iterations. But add Hessian generated noise every iteration. And to the same reason, we simplily set the coefficent of Hessian noise to $\sqrt{\mathrm{Tr}(\boldsymbol{H})/m\mathrm{Tr}(\boldsymbol{V})}$, to avoid extensively tuning of hyperparameter.
- **SGD**: Learning rate $\eta = 0.07$, mini batch size $m = 20$.

Estimation of Sharpness The sharpness are estimated by

$$\frac{1}{M}\sum_{j=1}^{M} L(\boldsymbol{\theta} + \boldsymbol{\xi}_j) - L(\boldsymbol{\theta}), \qquad \boldsymbol{\xi}_j \in \mathcal{N}(0, \sigma^2 I),$$

with $M = 1,000$ and $\sigma = 0.01$. These experiments are implemented by TensorFlow 1.5.0.

Projection residual For linearly independent column vector groups $A, B \in \mathcal{K}^{n\times k}, k << n$, the projection operator project A to B is defined as,

$$P_B(A) = B(B^TB)^{-1}B^TA. \tag{18}$$

Thus the projection residual can be measured by,

$$e(A, B) = \|A - P_B(A)\|_F^2. \tag{19}$$

When A, B are both column orthogonal, this equation can be simplified to,

$$e(A, B) = \left\|A - BB^TA\right\|_F^2, \tag{20}$$

and in this case, it is easily to show that $e(A, B)$ is commutative and $e(A, B) \in [0, k]$.

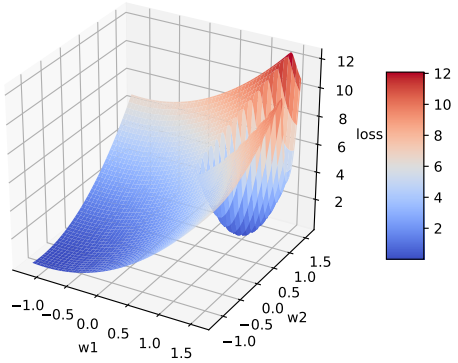


Figure 10. Constructed 2-dimensional surface in main paper.

D. Large Networks with Clean Data

Dataset Standard CIFAR-10 dataset without data augmentation.

Model Standard VGG11 network without any regularizations including dropout, batch normalization, weight decay, etc. The total number of parameters of this network is $9,750,922$.

Training details Learning rates $\eta_t = 0.05$ are fixed for all optimizers, which is tuned for the best generalization performance of **GD**. The batch size of **SGD** is $m = 100$. The noise std of **GLD constant** is $\sigma = 10^{-3}$.

Estimation of Sharpness The sharpness are estimated by

$$\frac{1}{M}\sum_{j=1}^{M} L(\boldsymbol{\theta} + \boldsymbol{\xi}_j) - L(\boldsymbol{\theta}), \qquad \boldsymbol{\xi}_j \in \mathcal{N}(0, \sigma^2 I),$$

with $M = 1,00$ and $\sigma = 0.01$.

These experiments are implemented by PyTorch 0.3.0.

E. Additional Experiments Evaluating the Significance of Noise

Figure 11 gives the dynamic of gradient mean and expected norm of noise during training using SGD. The dataset and model are same as the experiments of small networks with noisy data .

F. RMSProp diffusion experiments

The dataset, model architecture and estimation of sharpness are same as the experiments of small networks with noisy data .

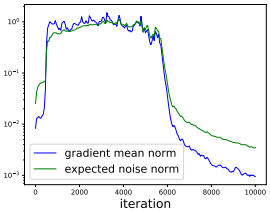


Figure 11. L_2 norm of gradient mean, $\|\boldsymbol{g}_0(\boldsymbol{\theta}_t)\|$, and expected norm of noise $\sqrt{\eta_t \mathbb{E}[\boldsymbol{\epsilon}_t^T \boldsymbol{\epsilon}_t]/m}$ during the training using SGD. The dataset is FashionMNIST with noisy label. The model is a LeNet-like CNN.

- **RMSProp**: Learning rate $\eta = 10^{-3}$, tuned in range of $\{0.5, 0.2, 0.1, 5 \times 10^{-2}, 10^{-2}, 5 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$. $\alpha = 0.99, \epsilon = 10^{-8}$, as PyTorch default setting, batch size $m = 20$.
- **SGD**: Learning rate $\eta = 0.1$, batch size $m = 20$.

Do to the moving average term of **RMSProp**, we cannot tune learning rate of **RMSProp** in diffusion stage. Thus we firstly tune **RMSProp** to the best generalization performance, then tune **SGD** initialized from this tuned **RMSProp**, to see if **SGD** can run out of the minima found by **RMSProp**.

These experiments are implemented by PyTorch 0.3.0.