On Linear Stability of SGD and Input-Smoothness of Neural Networks

Chao Ma
Department of Mathematics
Stanford University
Stanford, CA 94305
chaoma@stanford.edu

Lexing Ying
Department of Mathematics
Stanford University
Stanford, CA 94305
lexing@stanford.edu

Abstract

The multiplicative structure of parameters and input data in the first layer of neural networks is explored to build connection between the landscape of the loss function with respect to parameters and the landscape of the model function with respect to input data. By this connection, it is shown that flat minima regularize the gradient of the model function, which explains the good generalization performance of flat minima. Then, we go beyond the flatness and consider high-order moments of the gradient noise, and show that Stochastic Gradient Descent (SGD) tends to impose constraints on these moments by a linear stability analysis of SGD around global minima. Together with the multiplicative structure, we identify the Sobolev regularization effect of SGD, i.e. SGD regularizes the Sobolev seminorms of the model function with respect to the input data. Finally, bounds for generalization error and adversarial robustness are provided for solutions found by SGD under assumptions of the data distribution.

1 Introduction

Stochastic gradient descent (SGD) is the most widely used optimization algorithm to train neural networks [25, 4]. By taking mini-batches of training data instead of all the data in each iteration, it was firstly designed as a substitute of the gradient descent (GD) algorithm to reduce its computational cost. Extensive researches are conducted on the convergence of SGD, both on convex and non-convex objective functions [21, 22, 5, 3]. In these studies, convergence is usually proven in the cases where the learning rate is sufficiently small, hence the gradient noise is small. In practice, however, SGD is preferred over GD not only for the low computational cost, but also for the implicit regularization effect that produces solutions with good generalization performance [7, 16]. Since a trajectory of SGD tends to that of GD when the learning rate goes to 0, this implicit regularization effect must come from the gradient noise induced by mini-batch and a moderate learning rate.

When studying the gradient noise of SGD, a majority of work treat SGD as an SDE, and studies the noise of the SDE [15, 20, 17, 18, 32]. However, SGD is close to SDE only when the learning rate is small, and it is unclear that in practical setting whether the Gaussian noise of SDE can fully characterize the gradient noise of SGD. Some work resort to heavy-tailed noise like the Levy process [27, 37]. Another perspective to study the behavior of SGD is by its linear stability [30, 12]. This is relevant when the learning rate is not very small. The linear stability theory can explain the fast escape of SGD from sharp minima. The escape time derived from this theory depends on the logarithm of the barrier height, while the escape time derived from the diffusion theory based on SDE depends exponentially on the barrier height [32]. The former is more consistent with empirical observations [30].

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

An important observation that connects the generalization performance of the solution with the landscape of the loss function is that flat minima tend to generalize better [13, 31]. SGD is shown to pick flat minima, especially when the learning rate is big and the batch size is small [16, 15, 30]. Algorithms that prefer flat minima are designed to improve generalization [6, 14, 19, 35]. On the other hand, though, the reason why flat minima generalize better is still unclear. Intuitive explanations from description length or Bayesian perspective are provided in [13] and [23]. In [10], the authors show sharp minimum can also generalize by rescaling the parameters at a flat minimum. Hence, flatness is not a necessary condition of good generalization performance, but it is still possible to be a sufficient condition. In the study of linear stability in [30], besides the sharpness (a quantity inversely proportional to flatness), another quantity named non-uniformity is proposed which roughly characterizes the second order moment of the gradient noise. It is shown that SGD selects solutions with both low sharpness and low non-uniformity.

In this paper, we build a complete theoretical pipeline to analyze the implicit regularization effect and generalization performance of the solution found by SGD. Our starting points are the following two questions: (1) *Why SGD finds flat minima?* (2) *Why flat minima generalize better?* Our answers to these two questions go beyond the flatness and cover the non-uniformity and higher-order moments of the gradient noise. This distinguishes SGD from GD, and is out of the scope that can be explained by SDE. For the first question, we extend the linear stability theory of SGD from the second-order moments of the iterator of the linearized dynamics to the high-order moments. At the interpolation solutions found by SGD, by the linear stability theory, we derive a set of accurate upper bounds of the gradients' moment. For the second question, using the multiplicative structure of the input layer of neural networks, we show that the upper bounds obtained in the first step regularize the Sobolev seminorms of the model function. Finally, bridging the two components, our main result is a bound of generalization error under some assumptions of the data distribution. The bound works well when the distribution is supported on a low-dimensional manifold (or a union of low-dimensional manifolds). An informal statement of our main result is

(Main result) *Around an interpolation solution of the neural network model, assume that (1)the k-th order moment of SGD's iterator of the linearized dynamics is stable, (2)with probability at least $1-\varepsilon$ the testing data is close to a training data with distance smaller than δ, and (3) both the model function and the target function are upper bounded by a constant M. Then, at this interpolation solution we have*

$$\textit{generalization error} \lesssim n^{\frac{1}{k}}\delta^2 + \varepsilon M^2,$$

where n is the number of data. The bound depends on the learning rate and the batch size of SGD as constant factors.

The formal description is stated in Theorem 6 of Section 5. Our analysis also provide bounds for adversarial robustness. As a byproduct, we theoretically show that flatness of the minimum controls the gradient norm of the model function at the training data. Therefore, searching for flat minima has the effect of Lipschitz regularization, which is shown to be able to improve generalization [24].

Lying at the center of our analysis is the multiplicative structure of neural networks, i.e. in each layer the output features from the previous layer is multiplied with a parameter matrix. This structure is a rich source of implicit regularization. In this paper, we focus on the input layer, and build connection between the model function's derivative with respect to the parameters and its derivatives with respect to the data. Concretely, let $f(\mathbf{x}, W)$ be a neural network model, with $\mathbf{x}$ being the input data and W being the parameters. We split the parameters by $W = (W_1, W_2)$, where W_1 is the parameter matrix of the first layer, and W_2 denotes all other parameters. Then, the neural network can be represented by the form $f(\mathbf{x}, W) = \tilde{f}(W_1\mathbf{x}, W_2)$ due to the multiplicative structure of W_1 and $\mathbf{x}$. Accordingly $\nabla_{W_1}\tilde{f}(W_1\mathbf{x}, W_2)$ and $\nabla_{\mathbf{x}}\tilde{f}(W_1\mathbf{x}, W_2)$ take the form

$$\nabla_{W_1}\tilde{f}(W_1\mathbf{x}, W_2) = \frac{\partial f(W_1\mathbf{x}, W_2)}{\partial(W_1\mathbf{x})}\mathbf{x}^T, \qquad \nabla_{\mathbf{x}}\tilde{f}(W_1\mathbf{x}, W_2) = W_1^T\frac{\partial f(W_1\mathbf{x}, W_2)}{\partial(W_1\mathbf{x})}, \tag{1}$$

where $\frac{\partial f(W_1\mathbf{x}, W_2)}{\partial W_1\mathbf{x}}$ is usually a long expression produced by back propagation from the output to the first layer. By (1) we have

$$\|\nabla_{\mathbf{x}}\tilde{f}(W_1\mathbf{x}, W_2)\|_2 \leq \frac{\|W_1\|_2}{\|\mathbf{x}\|_2}\|\nabla_{W_1}\tilde{f}(W_1\mathbf{x}, W_2)\|_2. \tag{2}$$

Hence, $\nabla_{\mathbf{x}}\tilde{f}(W_1\mathbf{x}, W_2)$ is upper bounded by $\nabla_{W_1}\tilde{f}(W_1\mathbf{x}, W_2)$, given that W_1 is not too big and $\mathbf{x}$ is not too small. By the bound (2) we can derive the regularization effect of flatness at interpolation

solutions. To see this, let $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be the training data, and $\hat{L}(W) := \frac{1}{2n}\sum_{i=1}^n (f(\mathbf{x}_i, W) - y_i)^2$ be the empirical risk given by the square loss. Let $W^* = (W_1^*, W_2^*)$ be an interpolation solution, i.e. $f(\mathbf{x}_i, W^*) = y_i$ for any $i = 1, 2, ..., n$, and $\hat{L}(W^*) = 0$. Define the flatness of the minimum as the sum of the eigenvalues of $\nabla^2 \hat{L}(W^*)$, i.e. $\text{flatness}(W^*) = \text{Tr}(\nabla^2 \hat{L}(W^*))$. W^* being the interpolation solution implies

$$\text{flatness}(W^*) = \frac{1}{n}\sum_{i=1}^n \|\nabla_W f(\mathbf{x}_i, W^*)\|^2. \tag{3}$$

Hence, from (2) we can obtain the following equation that gives bounds for the gradients of the model function with respect to the input data at the training data,

$$\frac{1}{n}\sum_{i=1}^n \|\nabla_{\mathbf{x}} f(\mathbf{x}_i, W^*)\|^2 \leq \frac{\|W_1^*\|_2^2}{\min_i \|\mathbf{x}_i\|_2^2} \frac{1}{n}\sum_{i=1}^n \|\nabla_{W_1} f(\mathbf{x}_i, W^*)\|_2^2 \leq \frac{\|W_1^*\|_2^2}{\min_i \|\mathbf{x}_i\|^2} \text{flatness}(W^*). \tag{4}$$

The left hand side of (4) is usually used to regularize the Lipschitz constant of the model function, and such regularization can improve the generalization performance and adversarial robustness of the model. Hence, (4) reveals the regularization effect of flat minima. Later in the paper, we extend the analysis to higher-order moments of the gradient and combine the results with the linear stability theory of SGD to explain the implicit regularization effect of SGD. We also extend the bound on $\nabla_{\mathbf{x}} f(\mathbf{x}, W^*)$ from training data to all $\mathbf{x}$ in a neighborhood of the training data, which implies the regularization of flatness is actually stronger than the left-hand-side of (4).

The strong correlation between $\nabla_W f(\mathbf{x}, W)$ and $\nabla_{\mathbf{x}} f(\mathbf{x}, W)$ is justified by numerical experiments in practical settings. Specifically, in experiments we compare the following two quantities:

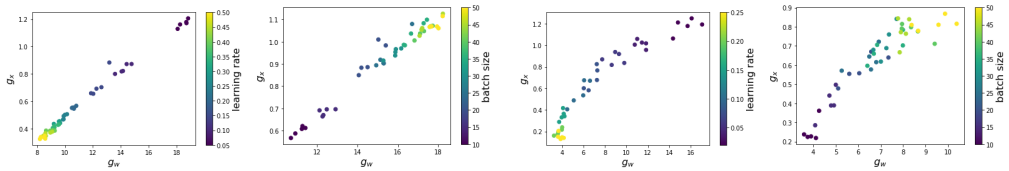
$$g_W := \left(\frac{1}{n}\sum_{i=1}^n \|\nabla_W f(\mathbf{x}_i, W)\|_2^2\right)^{\frac{1}{2}}, \quad g_{\mathbf{x}} := \left(\frac{1}{n}\sum_{i=1}^n \|\nabla_{\mathbf{x}} f(\mathbf{x}_i, W)\|_2^2\right)^{\frac{1}{2}}. \tag{5}$$

Figure 1 shows the results for a fully-connected network trained on FashionMNIST dataset and a VGG-11 network trained on CIFAR10 dataset. In each plot, g_W and $g_{\mathbf{x}}$ of different solutions found by SGD are shown by scatter plots. The plots show strong correlations between the two quantities. The colors of the points show that SGD with big learning rate and small batch size tends to converge to solutions with small g_W and $g_{\mathbf{x}}$, which is consistent with our theoretical results on the Sobolev regularization effect of SGD. To summarize, the main contributions of this paper are:

1. We extend the linear stability analysis of SGD to high-order moments of the iterators. At the solutions selected by SGD, we find a class of conditions satisfied by the gradients of different training data. These conditions cover the flatness and non-uniformity, and also include higher order moments. They characterize the regularization effect of SGD beyond GD and SDE.
2. By exploring the multiplicative structure of the neural networks' input layer, we build relations between the model function's derivatives with respect to the parameters and with respect to the inputs. By these relations we turn the conditions obtained for SGD into bounds of different Sobolev (semi)norms of the model function. In particular, we show that flatness of the minimum regularizes the L^2 norm of the gradient of the model function. This explains how the flatness (as well as other stability conditions for SGD) benefits generalization and adversarial robustness.
3. Still using the multiplicative structure, the bounds for Sobolev seminorms can be extended from the training data to a neighborhood around the training data, based on certain smoothness assumption of the model function (with respect to parameters). Then, bounds for generalization error and adversarial robustness are provided under reasonable assumptions of the data distribution. The bounds work well when the data are distributed effectively in a set that consists of low dimensional manifolds.

2 Preliminaries

Basic notations For any vector $\mathbf{x} = (x_1, ..., x_n)^T \in \mathbb{R}^n$ and any $p \geq 1$, $\|\mathbf{x}\|_p$ is the conventional p-norm of $\mathbf{x}$. If $g(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^n$ is a function with vector output, $\|g(\mathbf{x})\|_p$ means the p-norm of the vector $g(\mathbf{x})$, instead of a function norm. For any matrix $A \in \mathbb{R}^{m\times n}$, $\|A\|_p$ is the operator norm



(a) (b)

Figure 1: **(a)** A fully-connected neural network trained on FashionMNIST. **(b)** A VGG-11 network trained on CIFAR10. For both (a) and (b), the left panel shows g_W and $g_{\mathbf{x}}$ of the solutions found by SGD with different learning rate, while batch size fixed at 20. The right panel shows solutions found by SGD with different batch size, with learning rate fixed at 0.1.

induced by the p-norm on $\mathbb{R}^n$ and $\mathbb{R}^m$. $\|A\|_F$ denotes the Frobenius norm. Let $g: \mathbb{R}^n \to \mathbb{R}$ be a function, $\Omega \subset \mathbb{R}^n$ be a set in which g is defined, $q \in \mathbb{N}^*$, and $p \geq 1$. The Sobolev seminorm $|g|_{q,p,\Omega}$ is defined as $|g|_{q,p,\Omega} = (\sum_{|\alpha|=q} \int_\Omega |D^\alpha g(\mathbf{x})|^p d\mathbf{x})^{\frac{1}{p}}$, where $\alpha = (\alpha_1, ..., \alpha_n)$ is a multi-index with n positive integers, $|\alpha| = \sum_{i=1}^n \alpha_i$, and $D^\alpha g$ is defined as $D^\alpha g = \frac{\partial^{|\alpha|} g}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}}$. The index q can be extended to fractions, but in this paper we always consider integers. When $\Omega = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ is a finite set, we define the Sobolev seminorm as $|g|_{q,p,\Omega} = (\sum_{|\alpha|=q} \frac{1}{n} \sum_{i=1}^n |D^\alpha g(\mathbf{x}_i)|^p)^{\frac{1}{p}}$.

For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, we let $\Lambda(A)$ be the set of eigenvalues of A and $\lambda_{\max}(A)$ be the maximum eigenvalue of A. For two matrices A and B, $A \otimes B$ denotes the Kronecker product of A and B. For $k \in \mathbb{N}^*$, let $A^{\otimes k} = A \otimes A \otimes \cdots \otimes A$ where A is multiplied for k times. Therefore, if $A \in \mathbb{R}^{m\times n}$, then $A^{\otimes k} \in \mathbb{R}^{m^k \times n^k}$. However, for a vector $\mathbf{x} \in \mathbb{R}^n$, with an abuse of notation sometimes we also use $\mathbf{x}^{\otimes k}$ to denote the rank-one tensor product. In this case, $\mathbf{x}^{\otimes k} \in \mathbb{R}^{n\times n\times \cdots \times n}$.

Problem settings In this paper, we consider the learning of the parameterized model $f(\mathbf{x}, W)$, with $\mathbf{x}$ being the input data and W being the parameters. Let d be the dimension of the input data and w be the number of parameters in W, i.e. $\mathbf{x} \in \mathbb{R}^d$ and $W \in \mathbb{R}^w$. We assume that the model has scalar output. We consider the models with multiplicative structure on input data and part of the parameters. With an abuse of notation let $W = (W_1, W_2)$, where $W_1 \in \mathbb{R}^{m\times d}$ is the part of the parameters multiplied with $\mathbf{x}$ and W_2 denotes other parameters. Then, the model can be written as

$$f(\mathbf{x}, W) = \tilde{f}(W_1\mathbf{x}, W_2). \tag{6}$$

We remark that most neural networks have form (6), including convolutional networks, residual networks, recurrent networks, and even transformers. Note that since the $\mathbf{x}$ in (6) can also be understood as fixed features calculated using input data, (6) also includes random feature models.

We consider a supervised learning setting. Let μ be a data distribution supported within $\mathbb{R}^d$, and $f^*: \mathbb{R}^d \to \mathbb{R}$ be a target function. A set of n training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is obtained by sampling $\mathbf{x}_1, ..., \mathbf{x}_n$ i.i.d. from μ, and letting $y_i = f^*(\mathbf{x}_i)$. As mentioned in the introduction, the model is learned by minimizing the empirical loss function $\hat{L}(W) = \frac{1}{2n}\sum_{i=1}^n (f(\mathbf{x}_i, W) - y_i)^2$ in the space of parameters. The population loss, or the generalization error, is defined as $L(W) = \int (f(\mathbf{x}, W) - f^*(\mathbf{x}))^2 d\mu(\mathbf{x})$.

For any $i = 1, 2, ..., n$, let $L_i(W) = \frac{1}{2}(f(\mathbf{x}_i, W) - y_i)^2$ be the loss at $\mathbf{x}_i$. Then, the iteration scheme of the SGD with learning rate η and batch size B is

$$W_{t+1} = W_t - \frac{\eta}{B}\sum_{i=1}^{B} \nabla L_{\xi_i^t}(W_t), \tag{7}$$

where $\xi^t = (\xi_1^t, \xi_2^t, ..., \xi_B^t)$ is a B-dimensional random variable uniformly distributed on the B-tuples in $\{1, 2, ..., n\}$ and independent with W_t. In the paper, we study interpolation solutions W^* found by SGD, which satisfies $f(\mathbf{x}_i, W^*) = y_i$ for any $i = 1, 2, ..., n$. Obtaining interpolation solutions is possible in the over-parameterized setting [34], and is widely studied in existing work [36, 33].

Some assumptions will be made when deriving the generalization error bounds. Firstly, we assume the model function with respect to the parameter W to be smooth around W^*.

Definition 1. Let δ, C be positive numbers, and k be a positive integer. We say the model $f(\mathbf{x}, W)$ satisfies the (C, δ, k)**-local smoothness condition** at data $\mathbf{x}$ and parameter W, if for any W' such

that $\|W' - W\|_2 \leq \delta$, there is

$$\|\nabla_W f(\mathbf{x}, W')\|_{2k} \leq C(\|\nabla_W f(\mathbf{x}, W)\|_{2k} + 1). \tag{8}$$

Remark 1. *In the definition above, we consider the $2k$-norm of the gradients for the convenience of later analysis. When $k = 1$, the condition (8) becomes $\|\nabla_W f(\mathbf{x}, W')\|_2 \leq C(\|\nabla_W f(\mathbf{x}, W)\|_2 + 1)$. This is weaker than local approximation by Taylor expansion, which usually yields results like $\|\nabla_W f(\mathbf{x}_i, W) - \nabla_W f(\mathbf{x}_i, W^*)\|_2 \leq C\delta$. Here, we only require that the gradient with respect to the parameters does not get exceedingly large when W is close to W^*.*

Next, we need the data distribution μ to support roughly on a low dimensional manifold (or a union of low dimensional manifolds), thus the neighborhoods of training data can well cover all test data.

Definition 2. Let μ be a probability distribution supported in $\mathbb{R}^d$, and $\{\mathbf{x}_i\} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be a set of points in $\mathbb{R}^d$. For positive constants δ and ε, we say $\{\mathbf{x}_i\}$ (δ, ε)**-covers** μ if

$$\mathbb{P}_{\mathbf{x}\sim\mu}\left(\min_{1\leq i\leq n} \|\mathbf{x} - \mathbf{x}_i\|_2 > \delta\right) < \varepsilon, \tag{9}$$

i.e. with high probability a point sampled from μ lies close to a point in $\{\mathbf{x}_i\}$.

Definition 3. Let $\{\mathbf{x}_i\} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be a set of n points in $\mathbb{R}^d$. For positive constants δ and K, we say $\{\mathbf{x}_i\}$ is (δ, K)**-scattered** if the function $\kappa(\mathbf{x}) := \sum_{i=1}^n \mathbf{1}_{B(\mathbf{x}_i,\delta)}(\mathbf{x})$ satisfies $\kappa(\mathbf{x}) \leq K$. Here, $B(\mathbf{x}_i, \delta)$ is the ball in $\mathbb{R}^d$ centered at $\mathbf{x}_i$ with radius δ, and $\mathbf{1}_A(\mathbf{x})$ is the indicator function of set A.

Remark 2. *The uniform upper bound $\kappa(\mathbf{x}) \leq K$ in the above definition can be weakened as an integral upper bound $\frac{1}{V_{\mathcal{X}}}\int_{\mathcal{X}} \kappa(\mathbf{x})d\mathbf{x} \leq K$, where $\mathcal{X} = \bigcup_{i=1}^n B(\mathbf{x}_i, \delta)$ and $V_{\mathcal{X}}$ is the Lebesgue volume of $\mathcal{X}$.*

Definition 4. Let μ be a probability distribution supported in $\mathbb{R}^d$. For positive integer n and positive constants $\delta, \varepsilon_1, \varepsilon_2$, we say μ satisfies the $(n, \delta, \varepsilon_1, \varepsilon_2)$**-covered condition**, if with probability at least $1 - \varepsilon_1$ over the choice of n i.i.d. sampled data from μ, $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$, we have

$$\mathbb{P}_{\mathbf{x}\sim\mu}\left(\min_{1\leq i\leq n} \|\mathbf{x} - \mathbf{x}_i\|_2 > \delta\right) < \varepsilon_2. \tag{10}$$

On the other hand, for positive integer n and positive constants δ, ε_1 and K, we say μ satisfies the $(n, \delta, \varepsilon_1, K)$**-scattered condition**, if with probability at least $1 - \varepsilon_1$ over the choice of n i.i.d. sampled data $\{\mathbf{x}_i\}$ from μ, $\{\mathbf{x}_i\}$ is (δ, K)-scattered.

Later in the analysis of the generalization error (Section 5), we will assume the model and the data distribution satisfy the conditions in Definition 1 and 4 with appropriately chosen constants.

3 Linear stability theory of SGD

Compared with full-batch GD, SGD adds in each iteration a random "noise" to the gradient of the loss function. Hence, it is harder for SGD to be stable around minima than GD. This is shown by the linear stability theory of SGD.

Recall that the iteration scheme of SGD with learning rate η and batch size B is given by (7). Let W^* be an interpolation solution for the learning problem. When $\{W_t\}$ is close to W^*, the behavior of (7), including its stability around the minimum, can be characterized by the linearized dynamics at W^*:

$$\tilde{W}_{t+1} = \tilde{W}_t - \frac{\eta}{B}\sum_{i=1}^{B} \nabla_W f(\mathbf{x}_{\xi_i^t}, W^*)\nabla_W f(\mathbf{x}_{\xi_i^t}, W^*)^T(\tilde{W}_t - W^*). \tag{11}$$

The linearization is made by considering the quadratic approximation of L_i near W^*. For ease of notation, let $\boldsymbol{a}_i = \nabla_W f(\mathbf{x}_i, W^*)$, $H_i = \boldsymbol{a}_i\boldsymbol{a}_i^T$. In the linearized dynamics (11), without loss of generality we can set $W^* = 0$ by replacing $\tilde{W}_t$ by $\tilde{W}_t - W^*$. Then (11) becomes

$$\tilde{W}_{t+1} = \tilde{W}_t - \frac{\eta}{B}\sum_{i=1}^{B} \boldsymbol{a}_{\xi_i^t}\boldsymbol{a}_{\xi_i^t}^T\tilde{W}_t = \left(I - \frac{\eta}{B}\sum_{i=1}^{B} H_{\xi_i^t}\right)\tilde{W}_t. \tag{12}$$

Next, we define the stability of the above dynamics.

Definition 5. For any $k \in \mathbb{N}^*$, we say W^* is **k-th order linearly stable** for SGD with learning rate η and batch size B, if there exists a constant C (which may depend on k) that satisfies

$$\left\|\mathbb{E}\tilde{W}_t^{\otimes k}\right\|_F \leq C \left\|\mathbb{E}\tilde{W}_0^{\otimes k}\right\|_F,$$

for any $\tilde{W}_t$, $t \in \mathbb{N}^*$, given by the dynamics (12) from any initialization distribution of $\tilde{W}_0$.

$\{\tilde{W}_t\}$ is the trajectory of the linearized dynamics. It characterizes the performance of SGD around the minimum W^*, but does not equal to the original trajectory of SGD $\{W_t\}$. In practice, quadratic approximation of the loss function works in a neighborhood around the minimum. Hence, linear stability characterizes the local behavior of the iterators around the minimum. The $\tilde{W}_0$ in Definition 5 is the starting point of the linearized dynamics. It can be a point in the trajectory of the real dynamics, but it should not be understood as the initialization of the real dynamics (W_0). Since $\tilde{W}_t$ follows a linear dynamics, the $\tilde{W}_0$ here does not need to be close to W^*. Without linear stability of $\{\tilde{W}_t\}$, $\{W_t\}$ can never converge to the minimum. It is possible that $\{W_t\}$ oscillates around the minimum at a distance out of the reach of local quadratic approximation. However, empirical observations in [11] show that SGD usually converges to and oscillates in a region with good quadratic approximation. Hence, linear stability plays an important role in practice.

In [30], the following condition on the linear stability for $k = 2$ is provided.

Proposition 1. (Theorem 1 in [30]) *The global minimum W^* is 2nd-order linearly stable for SGD with learning rate η and batch size B if*

$$\lambda_{\max}\left\{(I - \eta H)^2 + \frac{\eta^2(n-B)}{B(n-1)}\Sigma\right\} \leq 1, \tag{13}$$

where $H = \frac{1}{n}\sum_{i=1}^{n} H_i$, and $\Sigma = \frac{1}{n}\sum_{i=1}^{n} H_i^2 - H^2$.

Shown by the proposition, the spectra of H and Σ influence the linear stability of SGD. Thus, in [30] the biggest eigenvalue of H and Σ are named sharpness and non-uniformity, respectively. The condition (13) is then be relaxed into a condition of sharpness and non-uniformity. Similar analysis was also conducted in [9] for linear least squares problems.

Remark 3. *The definition of stability in [30] concerns $\mathbb{E}\|\tilde{W}_t\|^2$, which is slightly different from the definition above when $k = 2$. However, since $\mathbb{E}\|\tilde{W}_t\|^2$ and $\|\mathbb{E}\tilde{W}_t^{\otimes 2}\|_F$ are equivalent, i.e. $\mathbb{E}\|\tilde{W}_t\|^2/\sqrt{w} \leq \|\mathbb{E}\tilde{W}_t^{\otimes 2}\|_F \leq \mathbb{E}\|\tilde{W}_t\|^2$, the two ways to define linear stability are equivalent.*

In the current work, we analyze the dynamics of $\mathbb{E}\tilde{W}_t^{\otimes k}$, which is a closed linear dynamics. As a comparison, the dynamics of $\mathbb{E}\|\tilde{W}_t\|^2$ is not closed, i.e. $\mathbb{E}\|\tilde{W}_{t+1}\|^2$ is not totally determined by $\mathbb{E}\|\tilde{W}_t\|^2$. By (12) we have

$$\tilde{W}_{t+1}^{\otimes k} = \left(I - \frac{\eta}{B}\sum_{i=1}^{B} H_{\xi_i^t}\right)^{\otimes k} \tilde{W}_t^{\otimes k}. \tag{14}$$

Let $\mathcal{I}_B$ be the set of all subsets of $\{1, 2, ..., n\}$ with cardinality B, and $\mathfrak{I} = \{i_1, ..., i_B\} \in \mathcal{I}_B$ be a batch. Note that ξ^t is independent with $\tilde{W}_t$. Taking expectation for (14) gives

$$\mathbb{E}\tilde{W}_{t+1}^{\otimes k} = \mathbb{E}\left(I - \frac{\eta}{B}\sum_{i=1}^{B} H_{\xi_i^t}\right)^{\otimes k} \mathbb{E}\tilde{W}_t^{\otimes k} = \frac{1}{\binom{n}{B}}\sum_{\mathfrak{I}\in\mathcal{I}}\left(I - \frac{\eta}{B}\sum_{j=1}^{B} H_{i_j}\right)^{\otimes k} \mathbb{E}\tilde{W}_t^{\otimes k} \tag{15}$$

Denote

$$T_{\mathfrak{I},k}^{\eta,B} := \left(I - \frac{\eta}{B}\sum_{j=1}^{B} H_{i_j}\right)^{\otimes k} \tag{16}$$

for each batch $\mathfrak{I}$, and $T_k^{\eta,B} = 1/\binom{n}{B}\sum_{\mathfrak{I}} T_{\mathfrak{I},k}^{\eta,B}$ be the expectation of $T_{\mathfrak{I}}^{\eta,B}$ over the choice of batches. Then we have $\mathbb{E}\tilde{W}_{t+1}^{\otimes k} = T_k^{\eta,B}\mathbb{E}\tilde{W}_t^{\otimes k}$.

On the other hand, if $\mathbb{E}\tilde{W}_t^{\otimes k}$ is understood as a tensor in $\mathbb{R}^{w\times w\times\cdots\times w}$, then it is always a symmetric tensor (any permutation of the index does not change the entry value). Hence, it has the decomposition $\mathbb{E}\tilde{W}_t^{\otimes k} = \sum_{i=1}^r \lambda_i \boldsymbol{v}_i^{\otimes k}$, with $\lambda_i \in \mathbb{R}$ and $v_i \in \mathbb{R}^w$ [8]. Let $\mathcal{M}_k$ be the set of symmetric tensors in $\mathbb{R}^{w\times w\times\cdots\times w}$, and $\mathcal{M}_k^+ = \{A \in \mathcal{M} :\ A = \sum_{i=1}^r \lambda_i v_i^{\otimes k}$ and $\lambda_i \geq 0$ for any $i = 1, ..., r\}$ be the set of "positive semidefinite" symmetric tensors. We provide the following conditions for the linear stability of SGD.

Theorem 1. *For any $k \in \mathbb{N}^*$, the global minimum W^* is k^{th} order linearly stable for SGD with learning rate η and batch size B, if and only if*

$$\|T_k^{\eta,B} A\|_F \leq \|A\|_F \tag{17}$$

holds for any $A \in \mathcal{M}_k^+$ if k is an even number, or for any $A \in \mathcal{M}_k$ is k is an odd number.

All proofs are provided in the appendix. As a corollary, when $k = 2$, we can write down another sufficient condition for linear stability than Proposition 1. See Corollary 3 in the appendix.

Recall that we let $\boldsymbol{a}_i = \nabla_W f(\mathbf{x}_i, W^*)$ and $H_i = \boldsymbol{a}_i \boldsymbol{a}_i^T$. The linear stability conditions in Theorem 1 imply the following bound on the moments of $\boldsymbol{a}_i$. Specifically, we have

Theorem 2. *If a global minimum W^* is k^{th} order linearly stable for SGD with learning rate η and batch size B, then, for any $j \in \{1, 2, ..., p\}$, we have*

$$\frac{1}{n}\sum_{i=1}^n \boldsymbol{a}_{i,j}^{2k} \leq \frac{2^k B^{k-1}}{\eta^k}, \tag{18}$$

where $\boldsymbol{a}_{i,j}$ means the jth entry of $\boldsymbol{a}_i$.

By Theorem 2, if a global minimum is stable for SGD with some order, then the gradient moment of this order is controlled. Summing j from 1 to w, we have $\frac{1}{n}\sum_{i=1}^n \|\nabla_W f(\mathbf{x}_i, W)\|_{2k}^{2k} \leq \frac{2w(2B)^{k-1}}{\eta^k}$. For $k = 1$ and 2, this gives control to the flatness and non-uniformity of the minimum, respectively. For general k, applying the Hölder inequality on (18), we have the following corollary which bounds the mean $2k$-norm of the gradients at the training data.

Corollary 1. *If a global minimum W^* is k^{th} order linearly stable for SGD with learning rate η and batch size B, then*

$$\frac{1}{n}\sum_{i=1}^n \|\nabla_W f(\mathbf{x}_i, W^*)\|_{2k} = \frac{1}{n}\sum_{i=1}^n \|\boldsymbol{a}_i\|_{2k} \leq \left(\frac{w}{B}\right)^{\frac{1}{2k}} \sqrt{\frac{2B}{\eta}}. \tag{19}$$

4 The Sobolev regularization effect

In this section, we build connection between $\nabla_W f(\mathbf{x}, W)$ and $\nabla_{\mathbf{x}} f(\mathbf{x}, W)$ using the multiplication of W_1 and $\mathbf{x}$. For general parameterized model, it is hard to build connection between the two gradients. However, this is possible for neural network models, in which the input variable is multiplied with a set of parameters (the first-layer parameter) before any non-linear operation. By this connection, at an interpolation solution that is stable for SGD, we can turn the moments bounds derived in the previous section into the bounds on $\nabla_{\mathbf{x}} f(\mathbf{x}, W)$. For different k, the moment bound on $\nabla_W f(\mathbf{x}, W)$ controls the Sobolev seminorms of $f(\cdot, W)$ at the training data with different index p. When $k = 1$, this becomes a bound for the sum of gradient square at the training data, which is used in the literature as a regularization term to regularize the Lipschitz constant of the model function. This explains the favorable generalization performance of flat minima.

Recall $f(\mathbf{x}, W) = \tilde{f}(W_1\mathbf{x}, W_2)$, and we can express $\nabla_{\mathbf{x}}\tilde{f}$ and $\nabla_{W_1}\tilde{f}$ as (1). Then, we easily have the following proposition.

Proposition 2. *Consider the model $\tilde{f}(W_1\mathbf{x}, W_2)$. For any $k \in \mathbb{N}^*$, any $W = (W_1, W_2)$ and $\mathbf{x} \neq 0$,*

$$\|\nabla_{\mathbf{x}}\tilde{f}(W_1\mathbf{x}, W_2)\|_{2k}^{2k} \leq \frac{\|W_1^T\|_{2k}^{2k}}{\|\mathbf{x}\|_{2k}^{2k}} \|\nabla_W \tilde{f}(W_1\mathbf{x}, W_2)\|_{2k}^{2k}, \tag{20}$$

As a corollary, we have the following bound for the Sobolev seminorm using $\nabla_W \tilde{f}$.

Corollary 2. *Let $f(\mathbf{x}, W) = \tilde{f}(W_1\mathbf{x}, W_2)$, and $\{\mathbf{x}_i\} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. Let $|f(\cdot, W)|_{1,2k,\{\mathbf{x}_i\}}$ be the Sobolev seminorm of $f(\cdot, W)$ on $\{\mathbf{x}_i\}$ as defined at the beginning of Section 2. Then,*

$$|f(\cdot, W)|_{1,2k,\{\mathbf{x}_i\}} \leq \frac{\|W_1^T\|_{2k}}{\min_{1\leq i\leq n} \|\mathbf{x}_i\|_{2k}} \left(\frac{1}{n} \sum_{i=1}^{n} \|\nabla_W f(\mathbf{x}_i, W)\|_{2k}^{2k} \right)^{\frac{1}{2k}}. \tag{21}$$

Combining Corollary 2 with the stability condition in Theorem 2, we have the following control for the Sobolev seminorm of the model function at interpolation solutions that are stable for SGD.

Theorem 3. *If a global minimum W^* which interpolates the training data is k^{th} order linearly stable for SGD with learning rate η and batch size B, and $W^* = (W_1^*, W_2^*)$. Then,*

$$|f(\cdot, W^*)|_{1,2k,\{\mathbf{x}_i\}} \leq \frac{\|(W_1^*)^T\|_{2k}}{\min_{1\leq i\leq n} \|\mathbf{x}_i\|_{2k}} \left(\frac{w}{B}\right)^{\frac{1}{2k}} \sqrt{\frac{2B}{\eta}}. \tag{22}$$

By Corollary 2 and Theorem 3, as long as the data are not very small (in practical problems, the input data are usually normalized), and the input-layer parameters are not very large, the Sobolev seminorms of the model function (evaluated at the training data) is regularized by the linear stability of SGD. The regularization effect on Sobolev seminorm gets stronger for bigger learning rate and smaller batch size. When k is big, the dependence of the bound with p is negligible.

If the solution W^* satisfies the local smoothness condition defined in Definition 1, the control on $\nabla_{\mathbf{x}} \tilde{f}(W_1^*\mathbf{x}, W_2^*)$ can be extended to a neighborhood around the training data. Then, we will be able to control the "population" Sobolev seminorms. First, around a certain training data, we have the following estimate.

Proposition 3. *Assume the model $f(\mathbf{x}, W)$ satisfies (C, δ_{approx}, k)-local smoothness condition for some $C, \delta_{approx} > 0$ and $k \in \mathbb{N}^*$, at W^* and $\mathbf{x}_*$. Then, for any $\mathbf{x}$ that satisfies $\|\mathbf{x} - \mathbf{x}_*\|_2 \leq \delta_{approx}\|\mathbf{x}_*\|_2/\|W_1^*\|_2$, we have*

$$\|\nabla_{\mathbf{x}} f(\mathbf{x}, W^*)\|_{2k} \leq \frac{C\|(W_1^*)^T\|_{2k}}{\|\mathbf{x}_*\|_{2k}} \left(\|\nabla_W f(\mathbf{x}_*, W^*)\|_{2k} + 1\right). \tag{23}$$

Proposition 3 turns the local smoothness condition in the parameter space into a local smoothness condition in the data space. This is made possible by the multiplicative structure of the network's input layer. Specifically, since in the first layer W_1 and $\mathbf{x}$ are multiplied together, a perturbation of $\mathbf{x}$ can be turned to a perturbation of W_1 without changing the value of their product. See the proof of the proposition in Appendix B.2 for more detail.

By the results above, we can obtain the following theorem which estimates the Sobolev seminorm of the model function on a neighborhood of the training data.

Theorem 4. *Let $W^* = (W_1^*, W_2^*)$ be an interpolation solution that is k^{th} order linearly stable for SGD with learning rate η and batch size B. Let $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be the training data. Suppose the model $f(\mathbf{x}, W)$ satisfies (C, δ_{approx}, k)-local smoothness condition at W^* and x_i for any $i = 1, 2, ..., n$. Consider the set $\mathcal{X}_\delta := \bigcup_{i=1}^{n} \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_i\| \leq \delta\}$ with $\delta \leq \delta_{approx}\|\mathbf{x}_i\|_2/\|W_1^*\|_2$. Assume $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ satisfies (δ, K)-scattered condition as defined in Definition 3. Then, we have*

$$|f(\mathbf{x}, W^*)|_{1,2k,\mathcal{X}_\delta} \leq \frac{(KV_{\mathcal{X}_\delta})^{\frac{1}{2k}} 2C\|(W_1^*)^T\|_{2k}}{\min_i \|\mathbf{x}_i\|_{2k}} \left(\left(\frac{w}{B}\right)^{\frac{1}{2k}} \sqrt{\frac{2B}{\eta}} + 1 \right), \tag{24}$$

where $V_{\mathcal{X}_\delta}$ is the Lebesgue volume of $\mathcal{X}_\delta$.

By Theorem 4, the model function found by SGD is smooth in a neighborhood of the training data, given that the landscape of the model in the parameter space is smooth. When $k = 1$, the results implies that flatness regularizes $|\nabla_{\mathbf{x}} f(\mathbf{x}, W^*)|_{1,2,\mathcal{X}_\delta}$, which is stronger than the gradient square at training data.

Discussion on the tightness of the bounds The bounds in this section depend on the norm of W_1^*. This norm may be big, especially when homogeneous activation functions, such as ReLU, is used. In

iterations	0	100	1000	10000	20000	50000
$\|W_1\|_2$	1.032	1.034	1.218	2.011	2.257	2.270

Table 1: The average l_2 norm of the first-layer parameter matrix of a fully-connected neural network trained on FashionMNIST dataset.

the ReLU case, for example, we can simply scale the first layer parameters to be arbitrarily large and second layer small while not changing the model function.

However, in practice SGD does not find these solutions with unbalanced layers because the homogeneity of ReLU induces an invariant in parameters. Specifically, let W_l and W_{l-1} be the parameter matrices in the lth and $(l-1)^{\text{th}}$ layers, then $W_{l-1}^T W_{l-1} - W_l W_l^T$ does not change significantly throughout the training (it is actually fixed in the zero learning rate limit). This invariant keeps layers balanced and prevents the first layer parameters from getting big alone.

On the other hand, we conduct numerical experiments to check the norm of W_1 during training. Table 1 shows the average l_2-norm of W_1 for the same experiments in part (a) of Figure1. The model is a fully-connected neural network, and the dataset is FashionMNIST. Results show that the norm of W_1 increases at the beginning, but becomes stable afterwards. It will not keep increasing during the training.

5 Generalization error and adversarial robustness

Regularizing the smoothness of the model function can improve generalization performance and adversarial robustness. This is confirmed in many practical studies [24, 29, 26]. Intuitively, a function with small gradient changes slowly as the input data changes, hence when the test data is close to one of the training data, the prediction error on the test data will be small. In this section, going along this intuition, we provide theoretical analysis of the generalization performance and adversarial robustness based on the Sobolev regularization effect derived in the previous sections. Still consider the model $f(\mathbf{x}, W) = \tilde{f}(W_1\mathbf{x}, W_2)$, and an interpolation solution $W^* = (W_1^*, W_2^*)$ found by SGD. First, based on Proposition 3 and Theorem 2, we show the following theorem, which is similar to Theorem 4 but estimates the maximum value of $\|\nabla_{\mathbf{x}} f(\mathbf{x}, W^*)\|$ in $\mathcal{X}_\delta$.

Theorem 5. *Let $W* = (W_1^*, W_2^*)$ be an interpolation solution that is k^{th} order linearly stable for SGD with learning rate η and batch size B. Let $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be the training data. Suppose the model $f(\mathbf{x}, W)$ satisfies (C, δ_{approx}, k)-local smoothness condition at W^* and $\mathbf{x}_i$ for any $1 = 1, 2, ..., n$. Recall the definition of $\mathcal{X}_\delta$ in Theorem 4. Then, for any $\mathbf{x} \in \mathcal{X}_\delta$, we have*

$$\|\nabla_{\mathbf{x}} f(\mathbf{x}, W^*)\|_{2k} \leq \frac{C\|(W_1^*)^T\|_{2k}}{\min_i \|\mathbf{x}_i\|_{2k}} \left(\left(\frac{2nw}{B} \right)^{\frac{1}{2k}} \sqrt{\frac{2B}{\eta}} + 1 \right). \tag{25}$$

With the result above, we can bound the generalization error if most test data are close to the training data, i.e. the data distribution μ satisfies a covered condition. This happens for machine learning problems with sufficient training data, especially for those where the training data lie approximately on a union of some low-dimensional surfaces. This is common in practical problems [1].

Theorem 6. *Suppose parameterized model $f(\mathbf{x}, W) = \tilde{f}(W_1\mathbf{x}, W_2)$ is used to solve a supervised learning problem with data distribution μ. Let f^* be the target function. Assume μ satisfies $(n, \delta, \varepsilon_1, \varepsilon_2)$-covered condition. Assume with probability no less than $1 - \varepsilon_1$ SGD with learning rate η and batch size B can find an interpolation solution W^* which is k^{th} order linearly stable, and (C, δ_{approx}, k)-local smoothness condition is satisfied at W^* and $\mathbf{x}_1, ..., \mathbf{x}_n$ with $\delta_{approx} \geq \delta\|W_1^*\|_2 / \min_i\{\|\mathbf{x}_i\|_2\}$. Further assume that both $|f(\cdot, W^*)|$ and $|f^*(\cdot)|$ are upper bounded by M_1, and $\|\nabla_{\mathbf{x}} f^*(\cdot)\|_2$ is upper bounded by M_2, for some constants M_1 and M_2. Then, with probability $1 - 2\varepsilon_1$ over the sampling of training data $\{\mathbf{x}_i\}_{i=1}^n$, we have*

$$\mathbb{E}_{\mathbf{x}\sim\mu}\|f(\mathbf{x}, W^*) - f^*(\mathbf{x})\|_2^2 \leq \frac{2dC^2\|(W_1^*)^T\|_{2k}^2}{\min_i \|\mathbf{x}_i\|_{2k}^2} \left(\left(\frac{2nw}{B} \right)^{\frac{1}{2k}} \sqrt{\frac{2B}{\eta}} + 1 \right)^2 \delta^2 + 2M_2^2\delta^2 + 4M_1^2\varepsilon_2. \tag{26}$$

The bound (26) tends to 0 as $n \to \infty$ as long as δ decays faster than $n^{-\frac{1}{2k}}$. From a geometric perspective, this happens when the dimension of the support of μ is less than k. And the lower the dimension, the faster the decay. It is possible to get rid of the n dependency in the bound when k is sufficiently large. We may use the estimate of Sobolev functions with scattered zeros, e.g. Theorem 4.1 in [2]. We leave the analysis to future work.

Adversarial robustness Neural network models suffer from adversarial examples [28], because the model function changes very fast in some directions so the function value becomes very different after a small perturbation. However, the results in Theorem 5 directly imply the adversarial robustness of the model at the training data. Hence, flatness, as well as high-order linear stability conditions, also imply adversarial robustness. Specifically, we have the following theorem.

Theorem 7. *Let $W* = (W_1^*, W_2^*)$ be an interpolation solution that is k^{th} order linearly stable for SGD with learning rate η and batch size B. Let $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be the training data. Suppose the model $f(\mathbf{x}, W)$ satisfies (C, δ_{approx}, k)-local smoothness condition at W^* and $\mathbf{x}_i$ for any $1 = 1, 2, ..., n$. Then, for any $\mathbf{x}$ that satisfies $\|\mathbf{x} - \mathbf{x}_i\|_2 \leq \delta$ for some $i \in \{1, ..., n\}$ and $\delta \leq \delta_{approx} \min_i \|\mathbf{x}_i\|_2 / \|W_1^*\|_2$, we have*

$$|f(\mathbf{x}, W^*) - f^*(\mathbf{x}_i, W^*)| \leq \frac{C\sqrt{d}\|(W_1^*)^T\|_{2k}}{\min_i \|\mathbf{x}_i\|_{2k}} \left(\left(\frac{2nw}{B} \right)^{\frac{1}{2k}} \sqrt{\frac{2B}{\eta}} + 1 \right) \delta. \tag{27}$$

6 Conclusion

In this paper, we connect the linear stability theory of SGD with the generalization performance and adversarial robustness of the neural network model. As a corollary, we provide theoretical insights of why flat minimum generalizes better. To achieve the goal, we explore the multiplicative structure of the neural network's input layer, and build connection between the model's gradient with respect to the parameters and the gradient with respect to the input data. We show that as long as the landscape on the parameter space is mild, the landscape of the model function with respect to the input data is also mild, hence the flatness (as well as higher order linear stability conditions) has the effect of Sobolev regularization. Our study reveals the significance of the multiplication structure between data (or features in intermediate layers) and parameters. It is an important source of implicit regularization of neural networks and deserves further exploration.

References

[1] Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *arXiv preprint arXiv:1905.12784*, 2019.

[2] Rémi Arcangéli, María Cruz López de Silanes, and Juan José Torrens. An extension of a bound for functions in sobolev spaces, with applications to (m, s)-spline interpolation and smoothing. *Numerische Mathematik*, 107(2):181–211, 2007.

[3] Raef Bassily, Mikhail Belkin, and Siyuan Ma. On exponential convergence of sgd in non-convex over-parametrized learning. *arXiv preprint arXiv:1811.02564*, 2018.

[4] Léon Bottou. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12, 1991.

[5] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[6] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

[7] Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE, 2018.

[8] Pierre Comon, Gene Golub, Lek-Heng Lim, and Bernard Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1254–1279, 2008.

[9] Alexandre Défossez and Francis Bach. Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Artificial Intelligence and Statistics*, pages 205–213. PMLR, 2015.

[10] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017.

[11] Yu Feng and Yuhai Tu. The inverse variance–flatness relation in stochastic gradient descent is critical for finding flat minima. *Proceedings of the National Academy of Sciences*, 118(9), 2021.

[12] Niv Giladi, Mor Shpigel Nacson, Elad Hoffer, and Daniel Soudry. At stability's edge: How to adjust hyperparameters to preserve minima selection in asynchronous training of neural networks? *arXiv preprint arXiv:1909.12340*, 2019.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

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

[15] Stanislaw Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

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

[17] Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pages 2101–2110. PMLR, 2017.

[18] Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. *J. Mach. Learn. Res.*, 20:40–1, 2019.

[19] Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*, 2018.

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

[21] Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in neural information processing systems*, 24:451–459, 2011.

[22] Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Advances in neural information processing systems*, 27:1017–1025, 2014.

[23] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

[24] Adam M Oberman and Jeff Calder. Lipschitz regularized deep neural networks converge and generalize. *arXiv preprint arXiv:1808.09540*, 2018.

[25] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[26] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[27] Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pages 5827–5837. PMLR, 2019.

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

[29] Dániel Varga, Adrián Csiszárik, and Zsolt Zombori. Gradient regularization improves accuracy of discriminative models. *arXiv preprint arXiv:1712.09936*, 2017.

[30] Lei Wu, Chao Ma, and Weinan E. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8289–8298, 2018.

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

[32] Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. *arXiv e-prints*, pages arXiv–2002, 2020.

[33] Zitong Yang, Yu Bai, and Song Mei. Exact gap between generalization error and uniform convergence in random feature models. *arXiv preprint arXiv:2103.04554*, 2021.

[34] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[35] Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation. *arXiv preprint arXiv:2010.04925*, 2020.

[36] Lijia Zhou, Danica J Sutherland, and Nathan Srebro. On uniform convergence and low-norm interpolation learning. *arXiv preprint arXiv:2006.05942*, 2020.

[37] Pan Zhou, Jiashi Feng, Chao Ma, Caiming Xiong, Steven HOI, et al. Towards theoretically understanding why sgd generalizes better than adam in deep learning. *arXiv preprint arXiv:2010.05627*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Our generalization bounds work on effectively low dimensional problems.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] In both Section 2 and the statement of theorems.
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section F and the URL therein.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Section F

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Section F

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proofs for Section 3

A.1 Proof of Theorem 1

In the proof, we ignore the superscripts η and B. We first show the sufficiency. Assume (17) holds. For any distribution of $\tilde{W}_0$, obviously we have $\mathbb{E}\tilde{W}_0^{\otimes k} \in \mathcal{M}_k$. Hence, if k is odd, linear stability comes directly from (17). If k is even, for any vector $\boldsymbol{v} \in \mathbb{R}^w$, we have

$$\mathbb{E}\tilde{W}_0^{\otimes k} \cdot \boldsymbol{v}^{\otimes k} = \mathbb{E}(\tilde{W}_0^T \boldsymbol{v})^k \geq 0,$$

which means $\mathbb{E}\tilde{W}_0^{\otimes k} \in \mathcal{M}_k^+$. Thus, the kth-order linear stability also holds for this distribution of $\tilde{W}_0$.

Next, we show the necessity. Let $A \in \mathcal{M}_k$, then A has the following decomposition

$$A = \sum_{i=1}^{r} \lambda_i \boldsymbol{v}_i^{\otimes k}$$

where $r \in \mathbb{N}^*$, λ_i are real numbers and $\boldsymbol{v}_i \in \mathbb{R}^w$. Then,

$$T_k A = \sum_{i=1}^{r} \lambda_i T_k(\boldsymbol{v}_i^{\otimes k}) = \frac{1}{\binom{n}{B}} \sum_{\mathfrak{I} \in \mathcal{I}} \sum_{i=1}^{r} \lambda_i \left(\left(I - \frac{\eta}{B} \sum_{j=1}^{B} H_{i_j} \right) \boldsymbol{v}_i \right)^{\otimes k}. \tag{28}$$

Hence, $T_k A$ is still symmetric, i.e. $T_k A \in \mathcal{M}_k$. Therefore, T_k induces a linear transform from $\mathcal{M}_k$ to $\mathcal{M}_k$. Let $\mathcal{T}_k$ be this linear transform. Since H_i is symmetric for all $i = 1, 2, ..., n$, if we understand $T_{\mathfrak{I}}$ as a matrix in $\mathbb{R}^{w^k \times w^k}$, then $T_{\mathfrak{I}}$ is symmetric for any batch $\mathfrak{I}$. Therefore, T_k is symmetric, which means $\mathcal{T}_k$ is also a symmetric linear transform. Then, we can easily show the following lemma by eigen-decomposition of $\mathcal{T}_k$:

Lemma 1. *For any $A \in \mathcal{M}_k$ and $A \neq 0$, if $\|\mathcal{T}_k A\|_F > \|A\|_F$, then $\lim_{m \to \infty} \|(\mathcal{T}_k)^m A\|_F = \infty$.*

The lemma is proven in Section D. With the lemma, the necessity follows by showing that we can find a distribution of $\tilde{W}_0$ such that $\mathbb{E}\tilde{W}_0^{\otimes k} = A$ for any $A \in \mathcal{M}_k^+$ if k is even and $A \in \mathcal{M}_k$ if k is odd. First consider an even k. For any $A \in \mathcal{M}_k^+$, we have the decomposition

$$A = \sum_{i=1}^{r} \lambda_i \boldsymbol{v}_i^{\otimes k}, \tag{29}$$

where $\lambda_i \geq 0$ for $i = 1, 2, ..., r$. Let the probability distribution of $\tilde{W}_0$ be given by the density function

$$p(W) := \sum_{i=1}^{r} \frac{\lambda_i}{\sum_{j=1}^{r} \lambda_j} \delta \left(W - (\sum_{j=1}^{r} \lambda_j)^{\frac{1}{k}} \boldsymbol{v}_i \right).$$

Then, we have $\mathbb{E}\tilde{W}_0^{\otimes k} = A$. Next, if k is odd, for any $A \in \mathcal{M}_k$, we still have decomposition (29), but some λ_i may be negative. However, since now k is an odd number, we can write the decomposition as

$$A = \sum_{i=1}^{r} |\lambda_i| (\text{sign}(\lambda_i) \boldsymbol{v}_i)^{\otimes k}.$$

Then, a similar construction as in the even case completes the proof.

A.2 Corollary 3 and the proof

Corollary 3. *The global minimum W^* is 2nd-order linearly stable for SGD with learning rate η and batch size B if*

$$\max \left| \lambda \left((I - \eta H)^{\otimes 2} + \frac{(n-B)}{B(n-1)} \frac{\eta^2}{n} \sum_{i=1}^{n} (H_i^{\otimes 2} - H^{\otimes 2}) \right) \right| \leq 1 \tag{30}$$

Proof:

When $k = 2$ we have

$$
\begin{aligned}
T_2^{\eta,B} &= \mathbb{E}_{\mathfrak{I}} \left(I - \frac{\eta}{B} \sum_{j=1}^{B} H_{i_j} \right)^{\otimes 2} \\
&= \mathbb{E}_{\mathfrak{I}} \left(I^{\otimes 2} - \frac{\eta}{B} \sum_{j=1}^{B} (I \otimes H_{i_j} + H_{i_j} \otimes I) + \frac{\eta^2}{B^2} \sum_{j_1, j_2 = 1}^{B} H_{i_{j_1}} \otimes H_{i_{j_2}} \right) \\
&= I^{\otimes 2} - \eta (I \otimes H + H \otimes I) + \frac{\eta^2}{B^2} \sum_{j_1, j_2 = 1}^{B} \mathbb{E}_{\mathfrak{I}} (H_{i_{j_1}} \otimes H_{i_{j_2}}). \qquad (31)
\end{aligned}
$$

For each $i \in \{1, 2, ..., n\}$, H_i appears in $\binom{n-1}{B-1}$ batches, and for each (i, j), $i, j \in \{1, 2, ..., n\}$, H_i and H_j appears in $\binom{n-2}{B-2}$ batches simultaneously. Hence,

$$
\begin{aligned}
\mathbb{E}_{\mathfrak{I}} \sum_{j=1}^{B} H_{i_j} \otimes H_{i_j} &= \sum_{i=1}^{n} \frac{\binom{n-1}{B-1}}{\binom{n}{B}} H_i \otimes H_i = \frac{B}{n} \sum_{i=1}^{n} H_i \otimes H_i \\
\mathbb{E}_{\mathfrak{I}} \sum_{j_1 \neq j_2} H_{i_{j_1}} \otimes H_{i_{j_2}} &= \sum_{i \neq j} \frac{\binom{n-2}{B-2}}{\binom{n}{B}} H_i \otimes H_j = \frac{B(B-1)}{n(n-1)} \sum_{i \neq j} H_i \otimes H_j.
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
\frac{\eta^2}{B^2} \sum_{j_1, j_2 = 1}^{B} \mathbb{E}_{\mathfrak{I}} (H_{i_{j_1}} \otimes H_{i_{j_2}}) &= \frac{\eta^2}{nB} \sum_{i=1}^{n} H_i \otimes H_i + \frac{\eta^2 (B-1)}{Bn(n-1)} \sum_{i \neq j} H_i \otimes H_j \\
&= \frac{\eta^2 (B-1)}{Bn(n-1)} \sum_{i,j=1}^{n} H_i \otimes H_j + \left(\frac{\eta^2}{nB} - \frac{\eta^2 (B-1)}{Bn(n-1)} \right) \sum_{i=1}^{n} H_i \otimes H_i \\
&= \frac{\eta^2 n (B-1)}{B(n-1)} H \otimes H + \frac{\eta^2 (n-B)}{Bn(n-1)} \sum_{i=1}^{n} H_i \otimes H_i \\
&= \eta^2 H \otimes H + \frac{(n-B)}{B(n-1)} \frac{\eta^2}{n} \sum_{i=1}^{n} (H_i^{\otimes 2} - H^{\otimes 2}). \qquad (32)
\end{aligned}
$$

Plug (32) into (31), we have

$$
T_2^{\eta,B} = (I - \eta H)^{\otimes 2} + \frac{(n-B)}{B(n-1)} \frac{\eta^2}{n} \sum_{i=1}^{n} (H_i^{\otimes 2} - H^{\otimes 2}). \qquad (33)
$$

Then, the result is a direct application of Theorem 1.

A.3 Proof of Theorem 2

By Theorem 1, for any $A \in \mathcal{M}_k^+$ we have $\|T_k^{\eta,B} A\|_F \leq \|A\|_F$. For any $j \in \{1, 2, ..., w\}$, let $\boldsymbol{e}_j \in \mathbb{R}^w$ be the jth unit coordinate vector, and let $A_j = \boldsymbol{e}_j^{\otimes k}$. Then $\|A\|_F = 1$. On the other hand,

$$
\begin{aligned}
(T_k^{\eta,B} A_j)_{j,j,...,j} &= \frac{1}{\binom{n}{B}} \sum_{\mathfrak{I}} (T_{\mathfrak{I},k}^{\eta,B} A_j)_{j,j,...,j} \\
&= \frac{1}{\binom{n}{B}} \sum_{\mathfrak{I}} \left(1 - \frac{\eta}{B} \sum_{k=1}^{B} \boldsymbol{a}_{i_k,j}^2 \right)^k
\end{aligned}
$$

Hence,

$$
\left| \frac{1}{\binom{n}{B}} \sum_{\mathfrak{I}} \left(1 - \frac{\eta}{B} \sum_{k=1}^{B} \boldsymbol{a}_{i_k,j}^2 \right)^k \right| \leq \|T_k^{\eta,B} A_j\|_F \leq 1. \qquad (34)
$$

Next, we will use the following lemma, whose proof is also provided in the appendix.

Lemma 2. *For any $t \geq 0$ and $k \in \mathbb{N}^*$, we have*

$$t^k \leq 2^{k-1}((t-1)^k + 1).$$

For any batch $\mathfrak{I} = \{i_1, i_2, ..., i_B\}$, let $t = \eta/B \sum_{k=1}^{B} \boldsymbol{a}_{i_k,j}^2$, we obtain

$$\left(\frac{\eta}{B}\sum_{k=1}^{B} \boldsymbol{a}_{i_k,j}^2\right)^k \leq 2^{k-1}\left(\left(\frac{\eta}{B}\sum_{k=1}^{B} \boldsymbol{a}_{i_k,j}^2 - 1\right)^k + 1\right).$$

Together with

$$\frac{\eta^k}{B^k}\sum_{k=1}^{B} \boldsymbol{a}_{i_k,j}^{2k} \leq \left(\frac{\eta}{B}\sum_{k=1}^{B} \boldsymbol{a}_{i_k,j}^2\right)^k,$$

we have

$$\frac{1}{B}\sum_{k=1}^{B} \boldsymbol{a}_{i_k,j}^{2k} \leq \frac{2^{k-1}B^{k-1}}{\eta^k}\left(\left(\frac{\eta}{B}\sum_{k=1}^{B} \boldsymbol{a}_{i_k,j}^2 - 1\right)^k + 1\right). \tag{35}$$

Taking expectation over batches, by (34) we have

$$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{a}_{i,j}^{2k} \leq \frac{(2B)^{k-1}}{\eta^k}\left(\frac{1}{\binom{n}{B}}\sum_{\mathfrak{I}}\left(\frac{\eta}{B}\sum_{k=1}^{B} \boldsymbol{a}_{i_k,j}^2 - 1\right)^k + 1\right) \leq \frac{2(2B)^{k-1}}{\eta^k}. \tag{36}$$

B Proofs for Section 4

B.1 Proof of Proposition 2

In this proof, $\|\cdot\|_{2k}$ always means the vector or matrix $2k$-norm, not the function norm. Then, we have

$$\|\nabla_{\mathbf{x}}\tilde{f}(W_1\mathbf{x}, W_2)\|_{2k}^{2k} = \left\|W_1^T \frac{\partial \tilde{f}(W_1\mathbf{x}, W_2)}{\partial(W_1\mathbf{x})}\right\|_{2k}^{2k} \leq \|W_1^T\|_{2k}^{2k}\left\|\frac{\partial \tilde{f}(W_1\mathbf{x}, W_2)}{\partial(W_1\mathbf{x})}\right\|_{2k}^{2k}.$$

On the other hand,

$$\sum_{j=1}^{m}\sum_{l=1}^{d}[\nabla_{W_1}\tilde{f}(W_1\mathbf{x}, W_2)]_{jl}^{2k} = \left\|\frac{\partial \tilde{f}(W_1\mathbf{x}, W_2)}{\partial(W_1\mathbf{x})}\right\|_{2k}^{2k}\|\mathbf{x}\|_{2k}^{2k}.$$

Hence,

$$\sum_{j=1}^{d}[\nabla_{\mathbf{x}}\tilde{f}(W_1\mathbf{x}, W_2)]_{j}^{2k} \leq \frac{\|W_1^T\|_{2k}^{2k}}{\|\mathbf{x}\|_{2k}^{2k}}\sum_{j=1}^{m}\sum_{l=1}^{d}[\nabla_{W_1}\tilde{f}(W_1\mathbf{x}, W_2)]_{jl}^{2k}.$$

Since W_1 is a subset of W, obviously we have

$$\sum_{j=1}^{m}\sum_{l=1}^{d}[\nabla_{W_1}\tilde{f}(W_1\mathbf{x}, W_2)]_{jl}^{2k} \leq \|\nabla_{W}\tilde{f}(W_1\mathbf{x}, W_2)\|_{2k}^{2k},$$

which completes the proof.

B.2 Proof of Proposition 3

Recall that $f(\mathbf{x}, W) = \tilde{f}(W_1\mathbf{x}, W_2)$. First, we find a W_1 such that $W_1\mathbf{x}_* = W_1^*\mathbf{x}$. Let $V = W_1 - W_1^*$, this is equivalent with solving the linear system

$$V\mathbf{x}_* = W_1^*(\mathbf{x} - \mathbf{x}_*)$$

for V. The linear system above is under-determined, hence solutions exist. We take the minimal norm solution

$$V = \frac{1}{\|\mathbf{x}_*\|_2^2}(W_1^*)^T(\mathbf{x}-\mathbf{x}_*)\mathbf{x}_*^T.$$

Especially, we have

$$\|W_1 - W_1^*\|_F = \|V\|_F = \frac{\|(W_1^*)^T(\mathbf{x}-\mathbf{x}_*)\|_2}{\|\mathbf{x}_*\|_2} \le \frac{\|W_1^*\|_2}{\|\mathbf{x}_*\|_2}\|\mathbf{x}-\mathbf{x}_*\|_2 \le \delta_{\text{approx}}.$$

Next, since $W_1\mathbf{x}_* = W_1^*\mathbf{x}$, we have $\tilde{f}(W_1^*\mathbf{x}, W_2^*) = \tilde{f}(W_1\mathbf{x}_*, W_2^*)$, and for gradient we have

$$\begin{aligned}
\|\nabla_{\mathbf{x}}\tilde{f}(W_1^*\mathbf{x}, W_2^*)\|_{2k} &\le \frac{\|(W_1^*)^T\|_{2k}}{\|\mathbf{x}\|_{2k}}\|\nabla_{W_1}\tilde{f}(W_1^*\mathbf{x}, W_2^*)\|_{2k} \\
&= \frac{\|(W_1^*)^T\|_{2k}}{\|\mathbf{x}\|_{2k}}\left\|\frac{\partial\tilde{f}(W_1^*\mathbf{x}, W_2^*)}{\partial(W_1^*\mathbf{x})}\right\|_{2k}\|\mathbf{x}\|_{2k} \\
&= \frac{\|(W_1^*)^T\|_{2k}}{\|\mathbf{x}\|_{2k}}\left\|\frac{\partial\tilde{f}(W_1\mathbf{x}_*, W_2^*)}{\partial(W_1^*\mathbf{x})}\right\|_{2k}\|\mathbf{x}\|_{2k}\frac{\|\mathbf{x}_*\|_{2k}}{\|\mathbf{x}_*\|_{2k}} \\
&= \frac{\|(W_1^*)^T\|_{2k}}{\|\mathbf{x}_*\|_{2k}}\|\nabla_{W_1}\tilde{f}(W_1\mathbf{x}_*, W_2^*)\|_{2k}. \qquad (37)
\end{aligned}$$

Let $W = (W_1, W_2^*)$, then $\|W - W^*\|_2 = \|W_1 - W_1^*\|_F \le \delta_{\text{approx}}$. Hence, by (8) we have

$$\|\nabla_{W_1}\tilde{f}(W_1\mathbf{x}_*, W_2^*)\|_{2k} \le \|\nabla_W\tilde{f}(W_1\mathbf{x}_*, W_2^*)\|_{2k} \le C\left(\|\nabla_W\tilde{f}(W_1^*\mathbf{x}_*, W_2^*)\|_{2k} + 1\right).$$

This together with (37) completes the proof of (23).

B.3 Proof of Theorem 4

Let $B(\mathbf{x}_i\delta)$ be the ball in $\mathbb{R}^d$ centered at $\mathbf{x}_i$ with radius δ. Then, for any $\mathbf{x} \in B(\mathbf{x}_i, \delta)$, by Proposition 3 we have

$$\|\nabla_{\mathbf{x}}f(\mathbf{x}, W^*)\|_{2k} \le \frac{C\|(W_1^*)^T\|_{2k}}{\|\mathbf{x}_i\|_{2k}}\left(\|\nabla_W f(\mathbf{x}_i, W^*)\|_{2k} + 1\right).$$

Hence,

$$\begin{aligned}
\int_{B(\mathbf{x}_i,\delta)}\|\nabla_{\mathbf{x}}f(\mathbf{x}, W^*)\|_{2k}^{2k}\mathbf{x} &\le V_{B_\delta}\frac{C^{2k}\|(W_1^*)^T\|_{2k}^{2k}}{\|\mathbf{x}_i\|_{2k}^{2k}}\left(\|\nabla_W f(\mathbf{x}_i, W^*)\|_{2k} + 1\right)^{2k} \\
&\le V_{B_\delta}\frac{2^{2k}C^{2k}\|(W_1^*)^T\|_{2k}^{2k}}{\|\mathbf{x}_i\|_{2k}^{2k}}\left(\|\nabla_W f(\mathbf{x}_i, W^*)\|_{2k}^{2k} + 1\right),
\end{aligned}$$

where V_{B_δ} is the volume of $B(\mathbf{x}_i, \delta)$, which does not depend on $\mathbf{x}_i$. Sum the above integral up for all training data, we have

$$\begin{aligned}
\int_{\mathcal{X}_\delta}\|\nabla_{\mathbf{x}}f(\mathbf{x}, W^*)\|_{2k}^{2k}\kappa(\mathbf{x})d\mathbf{x} &= \sum_{i=1}^n\int_{B(\mathbf{x}_i,\delta)}\|\nabla_{\mathbf{x}}f(\mathbf{x}, W^*)\|_{2k}^{2k}d\mathbf{x} \\
&\le V_{B_\delta}\frac{2^{2k}C^{2k}\|(W_1^*)^T\|_{2k}^{2k}}{\min_i\|\mathbf{x}_i\|_{2k}^{2k}}\left(\sum_{i=1}^n\|\nabla_W f(\mathbf{x}_i, W^*)\|_{2k}^{2k} + n\right) \\
&\le nV_{B_\delta}\frac{2^{2k}C^{2k}\|(W_1^*)^T\|_{2k}^{2k}}{\min_i\|\mathbf{x}_i\|_{2k}^{2k}}\left(\frac{2w(2B)^{k-1}}{\eta^k} + 1\right). \qquad (38)
\end{aligned}$$

On the other hand,

$$\int_{\mathcal{X}_\delta}\|\nabla_{\mathbf{x}}f(\mathbf{x}, W^*)\|_{2k}^{2k}\kappa(\mathbf{x})d\mathbf{x} \ge \int_{\mathcal{X}_\delta}\|\nabla_{\mathbf{x}}f(\mathbf{x}, W^*)\|_{2k}^{2k}d\mathbf{x}.$$

Therefore,

$$\frac{1}{V_{\mathcal{X}_\delta}}\int_{\mathcal{X}_\delta}\|\nabla_{\mathbf{x}} f(\mathbf{x},W^*)\|_{2k}^{2k}d\mathbf{x} \le \frac{nV_{B_\delta}}{V_{\mathcal{X}_\delta}}\frac{2^{2k}C^{2k}\|(W_1^*)^T\|_{2k}^{2k}}{\min_i\|\mathbf{x}_i\|_{2k}^{2k}}\left(\frac{2w(2B^{k-1})}{\eta^k}+1\right)$$
$$= \frac{1}{V_{\mathcal{X}_\delta}}\int_{\mathcal{X}_\delta}\kappa(\mathbf{x})d\mathbf{x}\frac{2^{2k}C^{2k}\|(W_1^*)^T\|_{2k}^{2k}}{\min_i\|\mathbf{x}_i\|_{2k}^{2k}}\left(\frac{2w(2B)^{k-1}}{\eta^k}+1\right)$$
$$\le \frac{K2^{2k}C^{2k}\|(W_1^*)^T\|_{2k}^{2k}}{\min_i\|\mathbf{x}_i\|_{2k}^{2k}}\left(\frac{2w(2B)^{k-1}}{\eta^k}+1\right). \tag{39}$$

Finally, we have

$$|f(\mathbf{x},W^*)|_{1,2k,\mathcal{X}_\delta} \le \frac{(KV_{\mathcal{X}_\delta})^{\frac{1}{2k}}2C\|(W_1^*)^T\|_{2k}}{\min_i\|\mathbf{x}_i\|_{2k}}\left(\left(\frac{w}{B}\right)^{\frac{1}{2k}}\sqrt{\frac{2B}{\eta}}+1\right). \tag{40}$$

C Proofs for Section 5

C.1 Proof of Theorem 5

Recall we let $\boldsymbol{a}_i = \nabla_W f(\mathbf{x}_i, W^*) \in \mathbb{R}^w$. By Theorem 2, we have

$$\frac{1}{n}\sum_{i=1}^n \boldsymbol{a}_{i,j}^{2k} \le \frac{2^k B^{k-1}}{\eta^k},$$

for any $j = 1, 2, ..., w$. Therefore, for any j,

$$\max_{1\le i\le n}\boldsymbol{a}_{ij}^{2k} \le \frac{2^k B^{k-1}n}{\eta^k}.$$

Sum j from 1 to w, we obtain

$$\max_{1\le i\le n}\|\boldsymbol{a}_i\|_{2k}^{2k} \le \sum_{j=1}^w \max_{1\le i\le n}\boldsymbol{a}_{ij}^{2k} \le \frac{2^k B^{k-1}nw}{\eta^k}.$$

Hence, for any $i = 1, 2, ..., n$, $\|\nabla_W f(\mathbf{x}_i, W^*)\|_{2k} \le \left(\frac{2nw}{B}\right)^{\frac{1}{2k}}\sqrt{\frac{2B}{\eta}}$, which together with Proposition 3 finishes the proof.

C.2 Proof of Theorem 6

Let $\mathcal{X}_\delta = \bigcup_{i=1}^n \{\mathbf{x}: \|\mathbf{x}-\mathbf{x}_i\|_2 \le \delta\}$. Then, we have

$$\mathbb{E}\|f(\mathbf{x},W^*) - f^*(\mathbf{x})\|_2^2 = \mathbb{E}\left[\|f(\mathbf{x},W^*) - f^*(\mathbf{x})\|_2^2 | \mathbf{x}\in\mathcal{X}_\delta\right] + \mathbb{E}\left[\|f(\mathbf{x},W^*) - f^*(\mathbf{x})\|_2^2 | \mathbf{x}\notin\mathcal{X}_\delta\right]$$
$$\le \mathbb{E}\left[\|f(\mathbf{x},W^*) - f^*(\mathbf{x})\|_2^2 | \mathbf{x}\in\mathcal{X}_\delta\right] + \varepsilon_2(2M_1^2), \tag{41}$$

where to be short we ignored the subscript $\mathbf{x}\sim\mu$ for the expectations. For any $\mathbf{x}\in\mathcal{X}_\delta$, let $\mathbf{x}_*$ be a training data that satisfies $\|\mathbf{x}-\mathbf{x}_*\| \le \delta$. By Theorem 5, for any $\mathbf{x}\in\mathcal{X}_\delta$ we have

$$\|\nabla_{\mathbf{x}} f(\mathbf{x},W^*)\|_{2k} \le \frac{C\|(W_1^*)^T\|_{2k}}{\min_i\|\mathbf{x}_i\|_{2k}}\left(\left(\frac{2nw}{B}\right)^{\frac{1}{2k}}\sqrt{\frac{2B}{\eta}}+1\right). \tag{42}$$

Therefore, by Hölder inequality,

$$|f(\mathbf{x},W^*) - f^*(\mathbf{x})| \le |f(\mathbf{x}_*,W^*) - f^*(\mathbf{x}_*)| + \max_{\mathbf{x}'\in\mathcal{X}_\delta}\|\nabla_{\mathbf{x}} f(\mathbf{x}',W^*)\|_{2k}\|\mathbf{x}-\mathbf{x}_*\|_{\frac{2k}{2k-1}}$$
$$+ \max_{\mathbf{x}'\in\mathcal{X}_\delta}\|\nabla_{\mathbf{x}} f^*(\mathbf{x}')\|_2\|\mathbf{x}-\mathbf{x}_*\|_2$$
$$\le \frac{C\|(W_1^*)^T\|_{2k}}{\min_i\|\mathbf{x}_i\|_{2k}}\left(\left(\frac{2nw}{B}\right)^{\frac{1}{2k}}\sqrt{\frac{2B}{\eta}}+1\right)\sqrt{d}\delta + M_2\delta. \tag{43}$$

In the last line, the $\sqrt{d}\delta$ term comes from

$$\|\mathbf{x}-\mathbf{x}_*\|_{\frac{2k}{2k-1}} \leq \|\mathbf{x}-\mathbf{x}_*\|_2 d^{\frac{k-1}{2k-1}} \leq \sqrt{d}\delta.$$

Hence, we have

$$\begin{aligned}\mathbb{E}\left[\|f(\mathbf{x},W^*)-f^*(\mathbf{x})\|_2^2|\mathbf{x}\in\mathcal{X}_\delta\right] &\leq \left[\frac{C\|(W_1^*)^T\|_{2k}}{\min_i\|\mathbf{x}_i\|_{2k}}\left(\left(\frac{2nw}{B}\right)^{\frac{1}{2k}}\sqrt{\frac{2B}{\eta}}+1\right)\delta+M_2\delta\right]^2 \\ &\leq \frac{2dC^2\|(W_1^*)^T\|_{2k}^2}{\min_i\|\mathbf{x}_i\|_{2k}^2}\left(\left(\frac{2nw}{B}\right)^{\frac{1}{2k}}\sqrt{\frac{2B}{\eta}}+1\right)^2\delta^2+2M_2^2\delta^2.\end{aligned} \tag{44}$$

Inserting (44) to (41) yields the result.

D Additional proofs

D.1 Proof of Lemma 1

We show the following more general result.

Lemma 3. *Let $A\in\mathbb{R}^{n\times n}$ be a symmetric matrix, and $\mathbf{x}\in\mathbb{R}^n$ be a vector. Then, if $\|A\mathbf{x}\|_2>\|\mathbf{x}\|_2$, we have*

$$\lim_{m\to\infty}\|A^m\mathbf{x}\|_2=\infty.$$

The proof is a simple practice for linear algebra. Let $A=Q\Sigma Q^T$ be the eigenvalue decomposition of A, and let $\boldsymbol{y}=Q^T\mathbf{x}$. Then, for any $m\in\mathbb{N}^*$ we have

$$\|A^m\mathbf{x}\|_2^2=\|\Sigma^m\boldsymbol{y}\|_2^2=\sum_{i=1}^n\sigma_i^{2m}y_i^2,$$

where σ_i are eigenvalues of A. Hence, $\|A\mathbf{x}\|_2>\|\mathbf{x}\|_2$ means

$$\sum_{i=1}^n\sigma_i^2y_i^2>\sum_{i=1}^n y_i^2,$$

which means there exists $j\in\{1,2,...,n\}$ such that $y_j\neq 0$ and $\sigma_i^2>1$. Then,

$$\lim_{m\to\infty}\|A^m\mathbf{x}\|_2^2=\lim_{m\to\infty}\sum_{i=1}^n\sigma_i^{2m}y_i^2\geq\lim_{m\to\infty}\sigma_j^{2m}y_j^2=\infty.$$

D.2 Proof of Lemma 2

When $t\in[0,1)$, we have $t^k+(1-t)^k\leq(t+1-t)^k=1$. Hence,

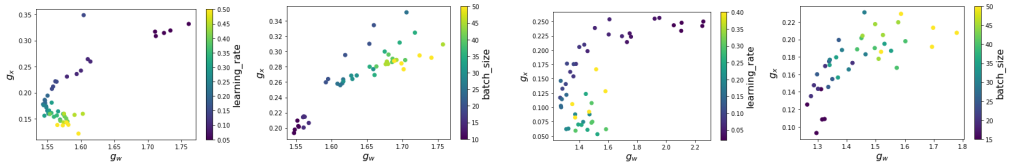
$$t^k\leq 1-(1-t)^k\leq 1+(t-1)^k\leq 2^{k-1}((t-1)^k+1).$$

When $t\geq 1$, by the Hölder inequality,

$$t=(t-1)+1\leq\left((t-1)^k+1\right)^{\frac{1}{k}}(1+1)^{1-\frac{1}{k}}.$$

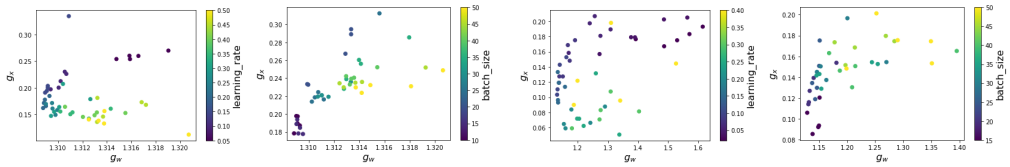
Taking k-th order on both sides, we have

$$t^k\leq 2^{k-1}((t-1)^k+1).$$



(a) (b)

Figure 2: Results for g_W^k and $g_{\mathbf{x}}^k$ with $k = 2$, on a fully-connected neural network trained on FashionMNIST (shown in (a)) and a VGG-11 network trained on CIFAR10 (shown in (b)). For both (a) and (b), the left panel shows g_W^k and $g_{\mathbf{x}}^k$ of the solutions found by SGD with different learning rate, while batch size fixed at 20. The right panel shows solutions found by SGD with different batch size, with learning rate fixed at 0.1.



(a) (b)

Figure 3: Results for g_W^k and $g_{\mathbf{x}}^k$ with $k = 3$, on a fully-connected neural network trained on FashionMNIST (shown in (a)) and a VGG-11 network trained on CIFAR10 (shown in (b)). For both (a) and (b), the left panel shows g_W^k and $g_{\mathbf{x}}^k$ of the solutions found by SGD with different learning rate, while batch size fixed at 20. The right panel shows solutions found by SGD with different batch size, with learning rate fixed at 0.1.

E Additional Experiments

Except the g_W and $g_{\mathbf{x}}$ in (5), we also checked the gradient norms with higher k in the same experiment settings. We consider

$$g_W^k := \left(\frac{1}{n} \sum_{i=1}^{n} \|\nabla_W f(\mathbf{x}_i, W)\|_{2k}^{2k} \right)^{\frac{1}{2k}}, \quad g_{\mathbf{x}}^k := \left(\frac{1}{n} \sum_{i=1}^{n} \|\nabla_{\mathbf{x}} f(\mathbf{x}_i, W)\|_{2k}^{2k} \right)^{\frac{1}{2k}}. \tag{45}$$

Figures 2 and 3 show the scatter plot for $k = 2$ and $k = 3$, respectively. The figures shows that in most cases there is still a strong correlation between the gradient norm with respect to W and the gradient norm with respect to $\mathbf{x}$.

The next figure (Figure 4) shows experiment results on more complicated dataset and network architectures, where we trained a 14-layer Resnet on a subset of the CIFAR100 dataset. The scattered plots show similar results as in Figrue 1, which further justifies our theoretical predictions.

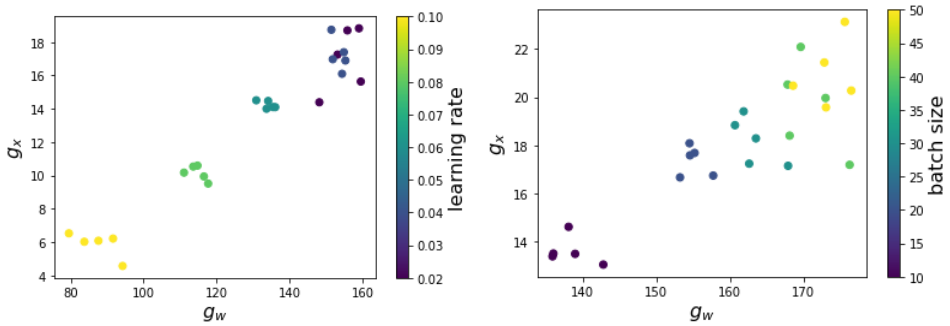


Figure 4: Results for a Resnet trained on CIFAR100 dataset. (Left) g_W and $g_{\mathbf{x}}$ of the solutions found by SGD with different learning rate, while batch size fixed at 20. (Right) g_W and $g_{\mathbf{x}}$ of the solutions found by SGD with different batch size, with learning rate fixed at 0.05.

F Experiment details

General settings In the numerical experiments shown by Figure 1, 2 and 3, we train fully-connected deep neural networks and VGG-like networks on FashionMNIST and CIFAR10, respectively. As shown in the figures, for each network, different learning rates and bacth sizes are chosen. 5 repetitions are conducted for each learning rate and batch size. In each experiment, SGD is used to optimize the network from a random initialization. The SGD is run for 100000 iterations to make sure finally the iterator is close to a global minimum. then, $g_{\mathbf{x}}$ and g_W in (5) are evaluated at the parameters given by the last iteration. In the experiments shown in Figure 4, we train a residual network on CIFAR100. Experiments are still repeated by 5 times in each combination of learning rate and batch size. In each experiment, SGD is run for 50000 iterations. All experiments are conducted on a MacBook pro 13" only using CPU. See the code at https://github.com/ChaoMa93/Sobolev-Reg-of-SGD.

Dataset For the FashionMNIST dataset, 5 out of the 10 classes are picked, and 1000 images are taken for each class. For the CIFAR10 dataset, the first 2 classes are picked with 1000 images per class. For the CIFAR100 dataset, the first 10 classes for picked with 500 images in each class.

Network structures The fully-connected network has 3 hidden layers, with 500 hidden neurons in each layer. The ReLU activation function is used. The VGG-like network consists of a series of convolution layers and max pooling layers. Each convolution layer has kernel size 3×3, and is followed by a ReLU activation function. The max poolings have stride 2. The order of the layers are

$$16->M->16->M->32->M->64->M->64->M,$$

where each number means a convolutional layer with the number being the number of channels, and "M" means a max pooling layer. A fully-connected layer with width 128 follows the last max pooling.

The residual network takes conventional architecture of Resnet. It consists of 6 residual blocks. The number of channels in the blocks are $32, 32, 64, 64, 128, 128$, from the input block to the output block.