On the Noisy Gradient Descent that Generalizes as SGD

Jingfeng Wu [1] **Wenqing Hu** [2] **Haoyi Xiong** [3] **Jun Huan** [4] **Vladimir Braverman** [1] **Zhanxing Zhu** [5]

Abstract

The gradient noise of SGD is considered to play a central role in the observed strong generalization abilities of deep learning. While past studies confirm that the magnitude and covariance structure of gradient noise are critical for regularization, it remains unclear whether or not the class of noise distributions is important. In this work we provide negative results by showing that noises in classes different from the SGD noise can also effectively regularize gradient descent. Our finding is based on a novel observation on the structure of the SGD noise: it is the multiplication of the gradient matrix and a sampling noise that arises from the mini-batch sampling procedure. Moreover, the sampling noises unify two kinds of gradient regularizing noises that belong to the Gaussian class: the one using (scaled) Fisher as covariance and the one using the gradient covariance of SGD as covariance. Finally, thanks to the flexibility of choosing noise class, an algorithm is proposed to perform noisy gradient descent that generalizes well, the variant of which even benefits large batch SGD training without hurting generalization.

1. Introduction

Stochastic gradient descent (SGD) is one of the standard workhorses for optimizing deep models (Bottou, 1991). Though initially proposed to remedy the computational bottleneck of gradient descent (GD), recent studies suggest SGD in addition induces a crucial implicit regularization, which prevents the over-parameterized models from converging to the minima that cannot generalize well (Zhang et al., 2017; Zhu et al., 2018; Jastrzębski et al., 2017; Hoffer et al., 2017; Keskar et al., 2017). To gain intuitions, one can compare the generalization abilities of (i) GD vs. SGD, (ii) small batch SGD vs. large batch SGD, and (iii) SGD vs. gradient Langevin dynamic (GLD). Empirical studies confirm that (i) SGD outperforms GD (Zhu et al., 2018), (ii) small batch SGD generalizes better than large batch SGD (Hoffer et al., 2017; Keskar et al., 2017), and (iii) GLD cannot compete with SGD (Zhu et al., 2018). To understand why these phenomena happen, let us look at the differences between the compared algorithms. Firstly SGD can be viewed as GD, an deterministic algorithm, with an unbiased noise inserted at every iteration, which is called the *gradient noise* (Bottou et al., 2018). Secondly the gradient noise of the small batch SGD has a much larger magnitude than that of the large batch SGD (Hoffer et al., 2017; Jastrzębski et al., 2017). Thirdly, even though the noise magnitude is tuned to be equal, the SGD noise has a nontrivial covariance structure, instead of just being a white noise as in GLD (Zhu et al., 2018). The above discussions exhibit a critical fact:

Certain noises can effectively regularize gradient descent.

Despite the efforts spent, this important yet implicit regularization effect induced by noise has never been fully understood. From the Bayesian perspective, the noise is interpreted to perform variational inference (Mandt et al., 2017; Chaudhari & Soatto, 2017). Such interpretation, however, requires unrealistic assumptions such as the noise has constant covariance (Mandt et al., 2017) or certain force is conservative (Chaudhari & Soatto, 2017). Another theory argues that the noise enables the gradient algorithm to escape from sharp minima (Zhu et al., 2018; Hu et al., 2019; Simsekli et al., 2019) that typically generalize worse (Hochreiter & Schmidhuber, 1997; Keskar et al., 2017). Hence GD enhanced by such noise tends to find flat minima that generalize well. This explanation hold valid to some extent; but the escaping behavior is too subtle to fit practice — the loss/accuracy does not jump significantly after the dynamic reaching a minimum, e.g., see the final epochs of Figure 4 in (Huang et al., 2017). Therefore the algorithm does not explicitly escape from minima in practice. Although the mechanism has not been completely understood, we can still recognize and utilize such implicit regularization by studying the properties of gradient noise.

We next summarize three important aspects of gradient noise that might introduce the regularization effects: noise magnitude, covariance structure and distribution class of noise.

[1]Johns Hopkins University, Baltimore, MD, USA [2]Missouri University of Science and Technology, Rolla, MO, USA [3]Big Data Laboratory, Baidu Research, Beijing, China [4]Styling.AI Inc., Beijing, China [5]Peking University, Beijing, China. Correspondence to: Jingfeng Wu <uuujf@jhu.edu>, Zhanxing Zhu <zhanxing.zhu@pku.edu.cn>.

Noise magnitude The large batch SGD encounters performance deterioration compared with the small batch one, thus the magnitude of gradient noise matters (Hoffer et al., 2017; Keskar et al., 2017). Furthermore, Jastrzębski et al. (2017) show that the ratio of learning rate to batch size, which directly controls the noise magnitude, has an important influence on the generalization of SGD: in a certain range, greater the ratio, larger the noise, and better the generalization.

Noise covariance structure From the perspective of escaping from minima, Zhu et al. (2018) emphasize the importance of the noise covariance structure for regularization. They show that when the noise covariance contains curvature information, it performs better for escaping from sharp minima (Zhu et al., 2018; Hu et al., 2019). Surprisingly, the covariance of the SGD noise aligns with the Hessian of the loss surface to some extent (Zhu et al., 2018; Li et al., 2019), which then partly explains the benefits brought by the SGD noise.

Noise class Many works assume that the SGD noise belongs to the Gaussian class due to the classical central limit theorem (Ahn et al., 2012; Chen et al., 2014; Shang et al., 2015; Mandt et al., 2017; Zhu et al., 2018). Nonetheless, Simsekli et al. (2019) first argue that the second moment of SGD noise might not exist, thus the Gaussianity assumption requires a second thought, since the classical central limit theorem has to be revised for heavy-tailed distributions (Gnedenko & Kolmogorov, 1968; Bertoin, 1998). Instead in this case, the central limit theorem leads to Levy distribution which they adopt for modeling SGD noise. By assuming so they obtain a faster escaping behavior of SGD (Simsekli et al., 2019; Nguyen et al., 2019; Şimşekli et al., 2019). Later Panigrahi et al. (2019) directly perform Gasussianity testing during the process of SGD learning deep neural networks. They empirically identify that when the batch size is greater than 256, the gradient noise of SGD can be treated as Gaussian in the early phase of training. But in general the SGD noise does not have to be Gaussian alike.

While past studies confirm the importance of noise magnitude and covariance structure, *the role of noise class in regularizing a gradient method has not been fully explored*. In this work, we attempt to address this issue from a novel perspective of sampling noise. Taking SGD for instance, we notice that the gradient noise is indeed caused by the mini-batch sampling procedure. This observation enables us to establish a key notion called the *sampling noise* to characterize the stochasticity of mini-batch sampling. Based on the sampling noise, we show that noises in classes different from the SGD noise can also effectively regularize gradient descent, thus provide negative evidence on the importance of the noise class. On the other hand, thanks to the flexibility of choosing noise class, we are allowed to use noisy gradient descent with best fitted noises based on practical requirements, beyond vanilla SGD. This finding supports the methods to employ structured Gaussian noises for improving GD/large batch SGD (Zhu et al., 2018; Wen et al., 2019). In summary, we obtain the following important results:

Contributions

1. A novel perspective is proposed for interpreting the SGD noise: it is the multiplication of the gradient matrix and a *sampling noise* which raises from the mini-batch sampling process. A general class of noisy gradient descent is thus defined based on the sampling noise.

2. The regularization role of the distribution class of gradient noise is then investigated. In both theory and experiments, we demonstrate that the noise class might not be a crux for regularization, provided suitable noise magnitude and covariance structure.

3. Two kinds of gradient regularizing noises from the Gaussian classes are then revised, i.e., the one using the (scaled) Fisher as covariance (Wen et al., 2019) and the one employing the gradient covariance of SGD as covariance (Zhu et al., 2018). The equivalence between them is established by analyzing their sampling noises.

4. Thanks to the unimportance of the noise class, an algorithm is proposed to perform generalizable noisy gradient descent with noises from various classes. Its variant even benefits large batch SGD training without hurting generalization.

2. The gradient noise of SGD

Let the training data be $\{x_i\}_{i=1}^n$, and consider the empirical loss $L(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(x_i;\theta)$, where $\ell(x;\theta)$ is the loss over one sample and $\theta \in \mathbb{R}^d$ is the parameter to be optimized. Define the *loss vector* as $\mathcal{L}(\theta) = \big(\ell(x_1;\theta),\dots,\ell(x_n;\theta)\big) \in \mathbb{R}^{1\times n}$, then the *gradient matrix* is $\nabla_\theta \mathcal{L}(\theta) = \big(\nabla_\theta \ell(x_1;\theta),\dots,\nabla_\theta \ell(x_n;\theta)\big) \in \mathbb{R}^{d\times n}$. Let $\mathbb{1} = (1,\dots,1)^T \in \mathbb{R}^n$, then $L(\theta) = \frac{1}{n}\mathcal{L}(\theta)\cdot \mathbb{1}$.

SGD During each iteration of SGD, the algorithm first randomly draws a mini-batch of samples with index set $B_t = \{i_1,\dots,i_b\}$ in size $|B_t| = b$, and then performs parameter update using the *stochastic gradient* $\tilde{g}(\theta)$ computed by the mini-batch and learning rate η,

$$\theta_{t+1} = \theta_t - \eta \tilde{g}(\theta_t), \quad \tilde{g}(\theta_t) = \frac{1}{b}\sum_{i\in B_t} \nabla_\theta \ell(x_i;\theta_t).$$

Sampling noise Note that the stochasticity of $\tilde{g}(\theta_t)$ is caused by the randomness of the mini-batch sampling procedure, thus the stochastic gradient could be written as

$$\tilde{g}(\theta_t) = \nabla_\theta \mathcal{L}(\theta_t) \cdot \mathcal{W}_{\text{sgd}},$$

where $\mathcal{W}_{\text{sgd}} \in \mathbb{R}^n$ is a random *sampling vector* characterizing the mini-batch sampling process. For instance considering mini-batch SGD without replacement, the sampling vector $\mathcal{W}_{\text{sgd}}$ contains exactly b multiples of $\frac{1}{b}$ and $n-b$ multiples of zero with random index. It is easy to see that $\mathbb{E}[\mathcal{W}_{\text{sgd}}] = \frac{1}{n}\mathbb{1}$, thus $\mathbb{E}[\tilde{g}(\theta_t)] = \frac{1}{n}\nabla_\theta \mathcal{L}(\theta_t) \cdot \mathbb{1} = \nabla_\theta L(\theta_t)$, i.e., the stochastic gradient $\tilde{g}(\theta_t)$ is an unbiased estimator of the full gradient $\nabla_\theta L(\theta_t)$.

Define the *sampling noise* as $\mathcal{V}_{\text{sgd}} = \mathcal{W}_{\text{sgd}} - \frac{1}{n}\mathbb{1}$. Then the stochastic gradient has the decomposition of

$$\tilde{g}(\theta_t) = \nabla_\theta L(\theta_t) + \nabla_\theta \mathcal{L}(\theta_t) \cdot \mathcal{V}_{\text{sgd}}, \quad \mathbb{E}[\mathcal{V}_{\text{sgd}}] = 0.$$

The first two moments of $\mathcal{V}_{\text{sgd}}$ are given in Proposition 1.

Proposition 1. *(Mean and covariance of the SGD sampling noise) For mini-batch sampled without replacement, the SGD sampling noise $\mathcal{V}_{sgd}$ satisfies*

$$\mathbb{E}[\mathcal{V}_{sgd}] = 0, \quad Var[\mathcal{V}_{sgd}] = \frac{n-b}{bn(n-1)}\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right).$$

For mini-batch sampled with replacement, the SGD sampling noise $\mathcal{V}'_{sgd}$ satisfies

$$\mathbb{E}[\mathcal{V}'_{sgd}] = 0, \quad Var[\mathcal{V}'_{sgd}] = \frac{1}{bn}\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right).$$

The proof is left in Section A.1 of the Supplementary Materials. If not stated otherwise, we focus on SGD with replacement in the remaining parts. However, our arguments hold for both of them with small modifications.

Gradient noise From the viewpoint of sampling noise, the *gradient noise* of SGD is the multiplication of the gradient matrix and its sampling noise,

$$\upsilon_{\text{sgd}}(\theta_t) = \tilde{g}(\theta_t) - \nabla_\theta L(\theta_t) = \nabla_\theta \mathcal{L}(\theta_t) \cdot \mathcal{V}_{\text{sgd}}.$$

Note that while the sampling noise $\mathcal{V}_{\text{sgd}}$ is independent with the parameter θ_t, the gradient noise $\upsilon_{\text{sgd}}(\theta_t)$ is coupled with the parameter θ_t. By Proposition 1, the first two moments of the gradient noise are $\mathbb{E}[\upsilon_{\text{sgd}}(\theta_t)] = \nabla_\theta \mathcal{L}(\theta_t)\mathbb{E}[\mathcal{V}_{\text{sgd}}] = 0$ and

$$\begin{aligned} C(\theta_t) &= \text{Var}[\upsilon_{\text{sgd}}(\theta_t)] = \nabla_\theta \mathcal{L}(\theta_t)\text{Var}[\mathcal{V}_{\text{sgd}}]\nabla_\theta \mathcal{L}(\theta_t)^T \\ &= \frac{1}{b}\left(\frac{1}{n}\nabla \mathcal{L}(\theta_t)\nabla \mathcal{L}(\theta_t)^T - \nabla L(\theta_t)\nabla L(\theta_t)^T\right). \end{aligned} \tag{1}$$

In the following we call $C(\theta_t)$ the *SGD covariance*.

As the structure of the SGD noise is clear, we turn to discuss the properties of the noise that affect its implicit regularization. Studies on large batch SGD training (Keskar et al., 2017; Hoffer et al., 2017) exhibit the importance of the *noise magnitude*, which is controlled by $\sqrt{\frac{\eta}{b}}$ (Jastrzębski et al., 2017). And from the viewpoint of escaping from minima, the implicit bias of SGD is also closely related to the *noise covariance structure* $C(\theta)$ (Zhu et al., 2018; Hu et al., 2019; Li et al., 2019). Recently, the role of the *noise class* raises research interests, as discussed below.

2.1. The class of the SGD noise

Due to the i.i.d. sampling of a mini-batch, as the batch size approaches infinity, the theory about limit theorems guarantees that the SGD noise converges to certain infinite divisible distribution (Gnedenko & Kolmogorov, 1968; Bertoin, 1998). If the second moment of the noise is finite, the limiting infinite divisible distribution will belong to the Gaussian class. Thus many works assume the Gaussianity of the SGD noise (Chen et al., 2014; Ahn et al., 2012; Shang et al., 2015; Mandt et al., 2017; Jastrzębski et al., 2017; Zhu et al., 2018). However, if the second moment does not exist, so that the noise is heavy-tailed, then the gradient noise should converge to a Levy type distribution, as assumed by (Simsekli et al., 2019; Nguyen et al., 2019; Şimşekli et al., 2019). Moreover, it is also questionable whether in practice the batch size is large enough for applying limit theorems. We investigate the two issues in the following.

The finiteness of the SGD covariance Based on analysis of the structure of the SGD noise, we have $C(\theta_t) = \nabla_\theta \mathcal{L}(\theta_t)\text{Var}[\mathcal{V}_{\text{sgd}}]\nabla_\theta \mathcal{L}(\theta_t)^T$ by Eq. (1), and $\text{Var}[\mathcal{V}_{\text{sgd}}]$ is finite by Proposition 1. Thus if the gradient matrix $\nabla_\theta \mathcal{L}(\theta_t)$ is bounded (almost everywhere), then $C(\theta_t)$ must be finite (almost everywhere). Firstly, the typical components of neural networks are twice differentiable (almost everywhere) (Goodfellow et al., 2016). Moreover, with the typical deep learning tricks such as near-zero initialization, early stopping, learning rate decay, weight decay, etc, the optimization process only happens in an small area around the near-zero initialization (Neyshabur et al., 2017; Jacot et al., 2018; Cao & Gu, 2019). Therefore the gradient matrix $\nabla_\theta \mathcal{L}(\theta_t)$ should be bounded almost everywhere in the area of our concerns. Thereby we argue that it is safe to assume the finiteness of the SGD covariance.

The non-Gaussianity of the SGD noise Even with finite covariance, it is still unclear whether in practice the batch size is sufficiently large for the Gaussian to be a good approximation for the SGD noise, especially when it comes to the extremely high dimensional parameter in deep learning. To validate this, Panigrahi et al. (2019) directly perform Gaussianity tests to the SGD noise during the training of

deep neural networks. They empirically find that when the batch size is greater than 256, the SGD noise behaves like a Gaussian one in the early phase of training. But generally the SGD noise does not belong to the Gaussian class.

The regularization role of the noise class We conclude that the SGD noise belongs to a particular distribution class that is neither Levy nor Gaussian. One might wonder if this particular distribution class of SGD noise is crucial for its regularization effects. In the remaining of this work, we address this issue by studying a general framework of noisy gradient descent which can employ noises from various classes, including the SGD noise class and the Gaussian class. The framework is called the *multiplicative SGD* (MSGD).

2.2. Multiplicative SGD

During each iteration, the proposed MSGD randomly generates a sampling vector $\mathcal{W} \in \mathbb{R}^n$ with mean as $\mathbb{E}[\mathcal{W}] = \frac{1}{n}\mathbb{1}$, and then takes update

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t)\mathcal{W}.$$

Denote the sampling noise as $\mathcal{V} = \mathcal{W} - \frac{1}{N}\mathbb{1}$, then the gradient noise is $\upsilon(\theta_t) = \nabla_\theta \mathcal{L}(\theta_t)\mathcal{V}$. Since our goal is to study the impact of noise class, the covariance of the gradient noise thus has to be fixed for excluding the influences of the noise magnitude and covariance structure. To this end it is sufficient to fix the covariance of the sampling noise, i.e., $\mathrm{Var}[\mathcal{V}] = \mathrm{Var}[\mathcal{V}_{\mathrm{sgd}}]$. The MSGD can then be written as

$$\begin{aligned} &\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta_t) + \eta \nabla_\theta \mathcal{L}(\theta_t)\mathcal{V}, \\ &\text{where} \quad \mathbb{E}[\mathcal{V}] = 0, \quad \mathrm{Var}[\mathcal{V}] = \mathrm{Var}[\mathcal{V}_{\mathrm{sgd}}]. \end{aligned} \tag{2}$$

In MSGD (2), the gradient noise $\upsilon(\theta_t)$ is decided by the deterministic gradient matrix $\nabla_\theta \mathcal{L}(\theta_t)$ and a sampling noise $\mathcal{V}$. Thus we can control the class of the gradient noise by choosing the class of the sampling noise. For example, the gradient noise $\upsilon(\theta_t)$ becomes the SGD noise if $\mathcal{V} = \mathcal{V}_{\mathrm{sgd}}$. Besides, if the sampling noise belongs to the Gaussian class, i.e., $\mathcal{V}_{\mathrm{G}} \sim \mathcal{N}\left(0, \mathrm{Var}[\mathcal{V}_{\mathrm{sgd}}]\right)$, then the gradient noise $\upsilon_{\mathrm{G}}(\theta_t) = \nabla_\theta \mathcal{L}(\theta_t)\mathcal{V}_{\mathrm{G}}$ is also Gaussian, i.e., $\upsilon_{\mathrm{G}}(\theta_t) \sim \mathcal{N}\left(0, C(\theta_t)\right)$, where $C(\theta_t) = \nabla_\theta \mathcal{L}(\theta_t)\mathrm{Var}[\mathcal{V}_{\mathrm{sgd}}]\nabla_\theta \mathcal{L}(\theta_t)^T$ by Eq. (1). In this case we call the iteration (2) the *Gaussian MSGD*. Moreover, gradient noises in other classes of practical interests can also be obtained with suitable sampling noises, e.g., Bernoulli sampling noises and sparse Gaussian sampling noises.

We then explore the role of the noise class by studying the generalization abilities of the MSGD (2) with noises from different classes.

3. Theoretical study

We first theoretically revise the role of the noise class for regularizing the algorithm. For the solution $\hat{\theta}$ found by noisy gradient descent and the optimal parameter θ_*, the generalization error can be measured as $\mathbb{E}_{x,\hat{\theta}}\left[\ell(x;\hat{\theta}) - \ell(x;\theta_*)\right]$. Now suppose the loss function $\ell(x;\theta)$ can be approximated by a quadratic one with respect to θ, then the generalization error involves just the first two moments of $\hat{\theta}$, which depends on at most the second moment information about the gradient noise, since the noise only accumulates linearly in the final solution $\hat{\theta}$ because of the linearity of the gradient. Hence intuitively, provided the noise covariance, the generalization error has little dependence on the particular class that the gradient noise belongs to.

To formalize the above intuition, we follow the setting of (Bach & Moulines, 2013; Dieuleveut et al., 2017; Défossez & Bach, 2015) and consider a online linear regression problem

$$\min_\theta f(\theta) := \frac{1}{2}\mathbb{E}_{(x,y)}[(x^T\theta - y)^2]. \tag{$\mathcal{P}$}$$

Let $\Sigma = \mathbb{E}_x[xx^T]$, then $f(\theta)$ always admits an optimal $\theta_* = \Sigma^\dagger \mathbb{E}_{(x,y)}[yx]$. Denote the residual as $\epsilon = y - x^T\theta_*$, then $\mathbb{E}[\epsilon x] = 0$. We adopt the following standard assumptions (Bach & Moulines, 2013; Dieuleveut et al., 2017; Défossez & Bach, 2015):

$$\mathbb{E}\left[\|x\|_2^2 xx^T\right] \preceq R^2\Sigma; \tag{$\mathcal{A}_1$}$$

$$\mathbb{E}\left[\epsilon^2 xx^T\right] \preceq \sigma^2\Sigma; \tag{$\mathcal{A}_2$}$$

$$\Sigma \preceq \lambda I. \tag{$\mathcal{A}_3$}$$

Remark. *The assumption ($\mathcal{A}_1$) is satisfied when the data is almost surely bounded, i.e., $\|x\|_2 \leq R$; and ($\mathcal{A}_2$) holds for almost surely bounded data or when the model is well-specified, i.e., ϵ_n is independent with x_n, and i.i.d. of zero mean and variance σ^2 (Dieuleveut et al., 2017).*

Typically the problem ($\mathcal{P}$) is learned by the averaged solution $\bar{\theta}_n = \frac{1}{n+1}\sum_{i=0}^n \theta_i$ of the (small batch) SGD (Bach & Moulines, 2013; Dieuleveut et al., 2017; Défossez & Bach, 2015)

$$\theta_{n+1} = \theta_n - \eta \sum_{r \in b_n} \left(x_r x_r^T \theta_n - y_r x_r\right), \tag{3}$$

where b_n is the index set of a randomly sampled mini-batch with a small batch size $|b_n| = b$. We note b could be 1. We consider the following (large batch) MSGD algorithm

$$\theta_{n+1} = \theta_n - \eta \sum_{r \in B_n} w_r \left(x_r x_r^T \theta_n - y_r x_r\right), \tag{4}$$

where B_n is the index set of a randomly sampled mini-batch, with a relatively large batch size $|B_n| = B > b$, and $\mathcal{W} = (w_{r_1}, \dots, w_{r_B})^T$ is a random sampling vector where $\mathbb{E}[\mathcal{W}] = \frac{1}{B}\mathbb{1}$.

The following theorem characterizes the generalization error of the large batch MSGD (4) and the small batch SGD (3).

Theorem 1. *Suppose the covariance of the sampling vector in MSGD (4) satisfies* $Var[\mathcal{W}] = \frac{B-b}{bB(B-1)}\left(I - \frac{1}{B}\mathbb{1}\mathbb{1}^T\right)$. *Then for both of the large batch MSGD (4) and the small batch SGD (3), we have*

$$\mathbb{E}_{\bar{\theta}_n}[f(\bar{\theta}_n)] - f(\theta_*) \leq \frac{C_1}{n+1} + \frac{C_2}{(n+1)^2},$$

where C_1 and C_2 are constants that depend on b, η, R, σ, λ and θ_0, but not B.

The proof is left in Supplementary Materials, Section A.2. The generalization error bound is indeed optimal as it matches the statistical lower bounds in certain circumstances (Dieuleveut et al., 2017).

According to Theorem 1, provided appropriate noise covariance (see Proposition 1), the large batch MSGD generalizes as the small batch SGD, and its generalization does not depend on the specific class of its gradient noise. Hence the noise class is not crucial for generalization, at least for the quadratic loss. For general loss functions, we empirically validate our understanding in the next section.

4. Empirical study

In this section we present our empirical results. The detailed setups of our experiments are explained in Supplementary Materials, Section C.

To begin with, we propose Algorithm 1 for efficiently performing the MSGD iteration (2). The key idea of Algorithm 1 is that the gradient operator commutes with the multiplication operator. Using Algorithm 1, we can easily inject noises with the SGD covariance to GD.

4.1. Gaussian noise with SGD covariance

In this part we discuss the ways to generate Gaussian gradient noises with covariance as the SGD covariance. Such noises in the Gaussian class is of great importance for both theoretical analysis of the implicit regularization (Zhu et al., 2018; Jastrzębski et al., 2017) and empirical algorithms for large batch SGD training (Wen et al., 2019). We denote the desired Gaussian noise as $\upsilon_{\mathrm{G}}(\theta) \sim \mathcal{N}\left(0, C(\theta)\right)$, where $C(\theta)$ is the SGD covariance as defined in Eq. (1).

SVD The typical approach of generating $\upsilon_{\mathrm{G}}(\theta)$ is based on the singular value decomposition (SVD) (Zhu et al., 2018). One first computes the covariance matrix and then applies SVD on it, $C(\theta) = U(\theta)\Lambda(\theta)U(\theta)^T$, then transforms a white noise $\epsilon \in \mathbb{R}^d$ into the Gaussian noise desired, $\upsilon_{\mathrm{G}}(\theta) = U(\theta)\Lambda(\theta)^{\frac{1}{2}}\epsilon$.

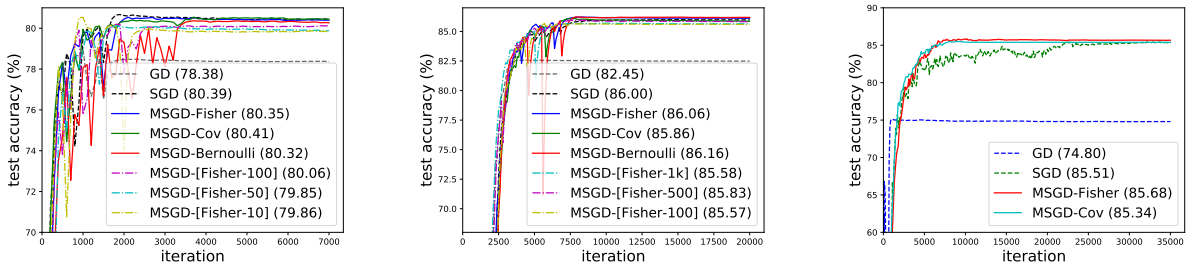
Algorithm 1 Multiplicative SGD

Input: Initial parameter $\theta_0 \in \mathbb{R}^d$, training data $\{(x_i, y_i)\}_{i=1}^n$, loss function $\ell_i(\theta) = \ell((x_i, y_i), \theta) \in \mathbb{R}$, loss vector $\mathcal{L}(\theta) = \left(\ell_1(\theta), \dots, \ell_n(\theta)\right) \in \mathbb{R}^{1\times n}$, learning rate $\eta > 0$
for $k = 0, 1, 2, ..., K-1$ **do**
  Generate a sampling noise $\mathcal{V} \in \mathbb{R}^n$ with zero mean and desired covariance
  Compute the sampling vector $\mathcal{W} = \frac{1}{n}\mathbb{1} + \mathcal{V}$
  Compute the randomized loss $\tilde{L}(\theta_k) = \mathcal{L}(\theta_k)\mathcal{W}$
  Compute the stochastic gradient $\nabla_\theta \tilde{L}(\theta_k)$
  Update the parameter $\theta_{k+1} = \theta_k - \eta\nabla_\theta \tilde{L}(\theta_k)$
end for
Output: Output θ_K

However, there are two obstacles in the above approach: (i) evaluating and storing the covariance matrix $C(\theta) \in \mathbb{R}^{d\times d}$ is computationally unacceptable, with both n and d being large; (ii) performing SVD for a $d \times d$ matrix is comprehensively hard when d is extremely large, e.g., deep neural networks. Furthermore, (i) and (ii) happen at every iteration of parameter update, since $C(\theta)$ depends on the parameter θ. In compromise, current works suggest to approximate $C(\theta)$ using only its diagonal or block diagonal elements (Wen et al., 2019; Zhu et al., 2018; Jastrzębski et al., 2017; Martens & Grosse, 2015). Generally, there is no guarantee that the diagonal information could approximate the full SGD covariance well; specifically, Zhu et al. (2018) demonstrate that such diagonal approximation cannot recover the regularization effects of SGD. Thus a more effective approach of generating Gaussian noise with the SGD covariance is demanded.

Gaussian sampling noise As discussed before, a gradient noise belongs to the Gaussian class if and only if its sampling noise is Gaussian. Thus based on the MSGD framework (2), to insert a Gaussian gradient noise $\upsilon_{\mathrm{G}}(\theta) \sim \mathcal{N}\left(0, C(\theta)\right)$, we only need to apply Algorithm 1 with its corresponding Gaussian sampling noise, which is $\mathcal{V}_{\mathrm{G}} \sim \mathcal{N}\left(0, \mathrm{Var}[\mathcal{V}_{\mathrm{sgd}}]\right)$ according to Eq. (1). Notice that the covariance of the SGD sampling noise admits a natural decomposition as $\mathrm{Var}[\mathcal{V}_{\mathrm{sgd}}] = \frac{1}{bn}\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right) = \frac{1}{bn}\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right)\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right)^T$. Thus the Gaussian sampling noise could be obtained by letting $\mathcal{V}_{\mathrm{G}} = \frac{1}{\sqrt{bn}}\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right)\epsilon$, where $\epsilon \in \mathbb{R}^n$ is a white noise. We use *MSGD-Cov* to name this approach of injecting Gaussian gradient noise with the SGD covariance.

Remark. *In the traditional setting of machine learning,*



(a) Small FashionMNIST (b) Small SVHN (c) CIFAR-10

Figure 1. The generalization of MSGD. X-axis: number of iterations; y-axis: test accuracy. **(a)**: We randomly draw $1,000$ samples from FashionMNIST as the training set, then train a small convolutional network with them. **(b)**: We use $25,000$ samples from SVHN as the training set, then train a VGG-11 without Batch Normalization. **(c)**: We train a ResNet-18 on CIFAR-10 without using data augmentation and weight decay. **MSGD-Fisher**: MSGD with Gaussian gradient noise whose covariance is the scaled Fisher. **MSGD-Cov**: MSGD with Gaussian gradient noise whose covariance is the SGD covariance. **MSGD-Bernoulli**: MSGD with Bernoulli sampling noise. **MSGD-[Fisher-B]**: MSGD-Fisher with the Fisher estimated using a mini-batch of samples in size B.

the number of samples is much larger than the number of parameters, $d \ll n$. And the SVD method for generating Gaussian noises is indeed plausible in this case. However, when it comes to deep neural networks where $n \ll d$, it turns out computing the full gradient could be much cheaper than explicitly evaluating the covariance matrix and performing SVD, resulting in the computational advantage of our approach over the traditional one.

Experiments In Figure 1 we test MSGD-Cov on various datasets and models. The results consistently suggest that the MSGD-Cov can generalize as the vanilla SGD, though its noise belongs to a different distribution class. More interestingly, we observe that the MSGD-Cov converges faster than the vanilla SGD.

4.2. Fisher vs. SGD covariance

In this part we discuss two kinds of commonly used Gaussian noises: the Gaussian noises with covariance as the SGD covariance, i.e., $\upsilon_{\mathrm{C}}(\theta) \sim \mathcal{N}(0, C(\theta))$ (Zhu et al., 2018) and the scaled Fisher, i.e., $\upsilon_{\mathrm{F}}(\theta) \sim \mathcal{N}(0, \frac{1}{b}F(\theta))$, where $F(\theta) = \frac{1}{n}\nabla_\theta \mathcal{L}(\theta)\nabla_\theta \mathcal{L}(\theta)^T$ is the Fisher. We call the MSGD with these two noises the MSGD-Cov and the *MSGD-Fisher*, respectively. The two noises sometimes cause confusion in literature, since both of them are adopted for simulating the SGD noise (Zhu et al., 2018; Wen et al., 2019), while we are not sure if they have the same effect. The connection between the SGD covariance and the Fisher is clear: $C(\theta) = \frac{1}{b}(F(\theta) - \nabla_\theta L(\theta)\nabla_\theta L(\theta)^T)$, i.e., ignoring a factor of scaling, $C(\theta)$ is the second central moment of the SGD noise, while $F(\theta)$ is the second raw moment. Nonetheless, their differences on imposing regularization have not been rigorously studied.

Intuitively the two dynamics should not be far away from each other. We can see this by investigating the MSGD iteration (2). At the early phase of the training, the gradient term is much larger than the noise term in scale (Shwartz-Ziv & Tishby, 2017) and dominates the optimization. Thus the noise term almost makes no contribution, no matter whether its covariance is the SGD covariance or the scaled Fisher. During the latter phase, however, the gradient turns to be close to zero, thus $C(\theta) \approx \frac{1}{b}F(\theta)$ and $\upsilon_{\mathrm{C}}(\theta) \approx \upsilon_{\mathrm{F}}(\theta)$. However, by such discussion neither the approximation is clear nor do we know about the transition phase.

By analyzing the sampling noises, we could develop a mathematical equivalence between the two noises along the whole training phase. Let $\mathcal{V}_{\mathrm{C}}$ and $\mathcal{V}_{\mathrm{F}}$ be the sampling noises for $\upsilon_{\mathrm{C}}(\theta)$ and $\upsilon_{\mathrm{F}}(\theta)$ respectively, i.e., $\upsilon_{\mathrm{C}}(\theta) = \nabla_\theta \mathcal{L}(\theta)\mathcal{V}_{\mathrm{C}}$ and $\upsilon_{\mathrm{F}}(\theta) = \nabla_\theta \mathcal{L}(\theta)\mathcal{V}_{\mathrm{F}}$. Then by the MSGD algorithm we have

$$\mathcal{V}_{\mathrm{C}} = \frac{1}{\sqrt{bn}}\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right)\epsilon = \frac{1}{\sqrt{bn}}\epsilon - \frac{1}{\sqrt{bn}}\frac{\mathbb{1}^T\epsilon}{n}\mathbb{1},$$
$$\mathcal{V}_{\mathrm{F}} = \frac{1}{\sqrt{bn}}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I_{n\times n}).$$

The matrix $\left(I - \frac{1}{N}\mathbb{1}\mathbb{1}^T\right)$ centralizes a random vector. But the components of the white noise ϵ are already i.i.d. of zero mean, thus $\frac{\mathbb{1}^T\epsilon}{n} \approx 0$ due to the law of large numbers. Hence $\mathcal{V}_{\mathrm{C}} \approx \mathcal{V}_{\mathrm{F}}$ and $\upsilon_{\mathrm{C}}(\theta) \approx \upsilon_{\mathrm{F}}(\theta)$. Moreover, the equivalence holds no matter where the parameter θ is, thus the Fisher Gaussian noise and the SGD covariance noise must lead to identical regularization effect for learning deep models.

Experiments In Figure 1 we present the experimental results regards MSGD-Cov and MSGD-Fisher. Consistent with our analysis, the behavior of the MSGD-Fisher perfectly approximates that of the MSGD-Cov. Hence the equivalence between the Fisher noise and the SGD covariance noise from the Gaussian class has been verified from

both theory and experiments. In the following study, we focus on MSGD-Fisher as the representative of the algorithms with noises from the Gaussian class.

4.3. Bernoulli sampling noise

Notice that the Fisher sampling noise $\mathcal{V}_{\rm F}$ has i.i.d. components and loses the covariance structure of the SGD sampling noise. Nonetheless it can still regularize GD well (see MSGD-Fisher in Figure 1). It seems that a sampling noise with independent components is capable enough for imposing regularization.

To further verify this conjecture, we consider a *Bernoulli sampling noise*: $\mathcal{V}_{\rm B} = (v_1, \dots, v_n)^T$, where the components are i.i.d. and $\mathbb{P}(v_i = \frac{1}{b}) = \frac{b}{n}, \mathbb{P}(w_i = 0) = \frac{n-b}{n}$. Then $\text{Var}[\mathcal{V}_{\rm B}] = \frac{n-b}{bn^2} I = \text{diag}\text{Var}[\mathcal{V}_{sgd}]$, i.e., the covariance of the Bernoulli sampling noise is exactly the diagonal of the covariance of SGD sampling noise. The Bernuolli sampling noise can also be easily injected to GD by Algorithm 1, and we call such algorithm the *MSGD-Bernoulli*.

Experiments In Figure 1 we find the MSGD-Bernoulli and the MSGD-Fisher both generalize as the vanilla SGD. In contrast, gradient noises with independent components can never recover the regularization effects of the SGD noise. For example one can look at the performance of *GLD diag* in (Zhu et al., 2018). This comparison reveals a fundamental advantage of understanding the gradient noise from its sampling noise.

4.4. Sparse Gaussian sampling noises

We then study another class of gradient noise who has *sparse Gaussian sampling noise*. The gradient noise is constructed as below: we first draw a mini-batch of samples uniformly at random in size B, then estimate the Fisher using this mini-batch, then generate a Gaussian noise using the estimated Fisher as covariance, finally the noise is properly scaled to maintain the magnitude. By Algorithm 1, the sampling noise is generated as

$$\mathcal{V}'_F = \sqrt{B/b} \cdot \mathcal{V}_{\rm sgd}(B) \odot \epsilon,$$

where $\mathcal{V}_{\rm sgd}(B)$ is the SGD sampling noise with batch size B, and $\epsilon \in \mathbb{R}^n$ is a white noise. We call the MSGD using $\mathcal{V}'_F$ as its sampling noise the *MSGD-[Fisher-B]*, where B denotes the batch size. Note that $\mathbb{E}[\mathcal{V}'_F] = 0$ and $\text{Var}[\mathcal{V}'_F] = \text{Var}[\mathcal{V}_{\rm sgd}(b)]$, i.e., the sampling noise has the same magnitude and covariance structure as the SGD sampling noise. Because $\mathcal{V}'_F$ is a sparse Gaussian noise, its gradient noise belongs to neither the Gaussian class nor the SGD noise class.

Experiments The performance of MSGD-[Fisher-B] is shown in Figures 1. Even with a very small batch size, e.g., 10 for FashionMNIST and 100 for SVHN, MSGD-[Fisher-B] can generalize as MSGD-Fisher and SGD. These results further support our understanding that the noise class is not the crux for regularization.

4.5. Mini-batch MSGD

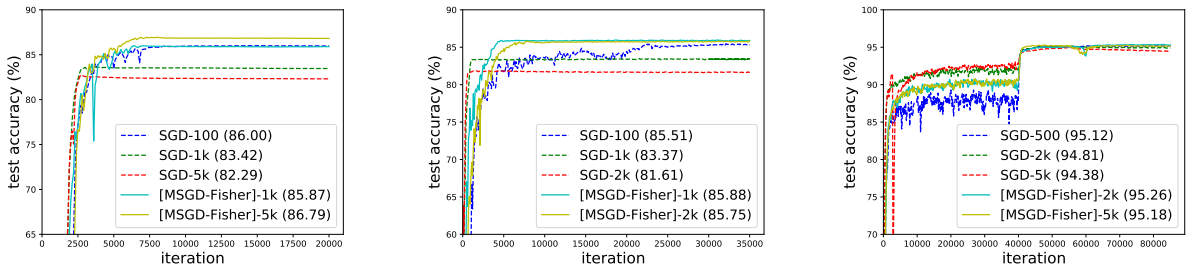
Finally, we discuss the mini-batch version of MSGD which is of practical interests. During each iteration of the vanilla MSGD, the information of full training set is required, which is unacceptable in practice. As an extension, we introduce Algorithm 2, the *mini-batch MSGD*. For example, when the plugged noise is Fisher Gaussian noise, we call the algorithm *[MSGD-Fisher]-B*, where B denotes the batch size of the mini-batch MSGD algorithm. We emphasize that the sampling noise in [MSGD-Fisher]-B is a sparse Gaussian noise plus an SGD sampling noise, thus it belongs to a new class different from what we have discussed before.

Algorithm 2 Mini-Batch Multiplicative SGD

Input: Initial parameter $\theta_0 \in \mathbb{R}^d$, training data $\{(x_i, y_i)\}_{i=1}^n$, loss function $\ell_i(\theta) = \ell((x_i, y_i), \theta) \in \mathbb{R}$, learning rate $\eta > 0$, batch size b
for $k = 0, 1, 2, ..., K-1$ **do**
  Uniformly sample a mini-batch $\{k_1, \dots, k_b\}$ and collect the loss vector $\mathcal{L}(\theta_k) = \big(\ell_{k_1}(\theta_k), \dots, \ell_{k_b}(\theta_k)\big) \in \mathbb{R}^{1\times b}$
  Generate a sampling noise $\mathcal{V} \in \mathbb{R}^b$ of zero mean and desired covariance
  Compute the sampling vector $\mathcal{W} = \frac{1}{b}\mathbb{1} + \mathcal{V}$
  Calculate the randomized loss $\tilde{L}(\theta_k) = \mathcal{L}(\theta_k)\mathcal{W}$
  Compute the stochastic gradient $\nabla_\theta \tilde{L}(\theta_k)$
  Update the parameter $\theta_{k+1} = \theta_k - \eta \nabla_\theta \tilde{L}(\theta_k)$
end for
Output: Output θ_K

Large batch training When training with SGD, as the batch size becomes large, the generalization gets hurt since the gradient noise tends to be small (Keskar et al., 2017). A promising method to close the generalization gap of large batch training is adding a compensatory gradient noise, e.g., a Gaussian gradient noise using scaled Fisher as covariance (Wen et al., 2019). However as we have discussed in Section 4.1, it is computationally costly to directly insert a structured Gaussian noise via SVD. Instead, the algorithm [MSGD-Fisher]-B provides an efficient method for injecting a compensatory sampling noise from the (sparse) Gaussian class.

Experiments We thus perform large batch training experiments with our [MSGD-Fisher]-B algorithm. The results are shown in Figure 2. Since in this case the covariance of the sampling noise becomes hard to calculate, we simply



(a) Small SVHN (b) CIFAR-10 (c) CIFAR-10 Standard

Figure 2. The generalization of mini-batch MSGD. X-axis: number of iterations; y-axis: test accuracy. **(a)**: We use $25,000$ samples from SVHN as the training set, then train a VGG-11 without Batch Normalization. **(b)**: We train a ResNet-18 on CIFAR-10 without using data augmentation and weight decay. **(c)**: We train a ResNet-18 on CIFAR-10 with full tricks. **SGD-B**: SGD with batch size B. **[MSGD-Fisher]-B**: mini-batch MSGD with batch size B, and an compensatory sampling noise from the (sparse) Gaussian class.

tune the noise magnitude to achieve its best performance. As illustrated in Figure 2 (a) (b), on toy datasets the [MSGD-Fisher]-B with large batch size has a even better generalization compared with small batch SGD. Its convergence is also faster. Even in real settings of training ResNet-18 on CIFAR10, Figure 2 (c) demonstrates that the [MSGD-Fisher]-B with large batch size generalizes well as the small batch SGD, while SGD with large batch size performs worse.

4.6. Empirical studies summary

In Figure 1 we compare the generalization performance of noisy gradient descents with noises from various different classes. We find that, provide suitable magnitude and covariance structure, all the concerned noises can regularize gradient descent as the SGD noise. These empirical results together with the theoretical evidence verify our understanding that the noise class is not a crux for regularization. An interesting additional finding is that Gaussian MSGD tends to converge faster than others.

In Figure 2 we present the empirical results of the mini-batch MSGD realized by Algorithm 2. Our algorithm perfectly closes the generalization gap of large batch training by injecting compensatory (sparse) Gaussian sampling noises. Besides, our algorithm achieves this effect in a more efficient manner than the traditional way of inserting Gaussian gradient noise based on SVD. These results demonstrate the promising application of the mini-batch MSGD algorithm in practice.

5. Discussion

Benefits of Gaussian gradient noise The continuous stochastic differential equations (SDEs) have been widely used for approximating and analyzing the discrete SGD iterations (Li et al., 2017; Hu et al., 2017; Orvieto & Lucchi, 2019). For SGD, this continuous approximation only hold in weak sense (Li et al., 2017). For Gaussian MSGD, however, a strong convergence can be established between the discrete iterations and the continuous SDEs. We provide a proof in Supplementary Materials, Section B. The strong convergence guarantees a path-wise closeness between the discrete iterations and the continuous paths, beyond the close behavior at the level of probability distributions guaranteed by weak convergence. This advantage of Gaussian MSGD might account for its observed faster convergence.

The importance of the gradient matrix $\nabla_\theta \mathcal{L}(\theta)$ Consider the MSGD-Bernoulli/Fisher and the *GLD diag* from (Zhu et al., 2018), empirical studies show that the MSGD-Bernoulli/Fisher generalize well as SGD, while the GLD diag performs much worse. In the MSGD-Bernoulli/Fisher the sampling noises have independent components, and in the GLD diag the gradient noise has independent components. Though the compared algorithms all discard certain "dependece" in their noises, the MSGD-Bernoulli/Fisher keep the full information of the gradient matrix, while the GLD diag severally destroys its structure. We thus conjecture that the gradient matrix contains key information for the regularization induced by noises.

6. Conclusion

In this work we introduce a novel kind of gradient noise as the composition of the gradient matrix and a sampling noise, which includes the SGD noise. By investigating these noises we find the noise class is not a crux for regularization, provided suitable noise magnitude and covariance structure. Furthermore, we show that the scaled Fisher and the gradient covariance of SGD is equivalent when serve as the covariance of noises from the Gaussian class. Finally, an algorithm is proposed to perform noisy gradient descent that generalizes as SGD. The algorithm can be extended for practical usage like large batch training.

References

Ahn, S., Korattikara, A., and Welling, M. Bayesian posterior sampling via stochastic gradient fisher scoring. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1771–1778. Omnipress, 2012.

Bach, F. and Moulines, E. Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n). In *Advances in neural information processing systems*, pp. 773–781, 2013.

Bertoin, J. *Levy Processes (Cambridge Tracts in Mathematics)*. Cambridge University Press, 1998.

Borkar, V. S. and Mitter, S. K. A strong approximation theorem for stochastic recursive algorithms. *Journal of optimization theory and applications*, 100(3):499–513, 1999.

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

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 10835–10845, 2019.

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

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, pp. 1683–1691, 2014.

Défossez, A. and Bach, F. Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Artificial Intelligence and Statistics*, pp. 205–213, 2015.

Dieuleveut, A., Flammarion, N., and Bach, F. Harder, better, faster, stronger convergence rates for least-squares regression. *The Journal of Machine Learning Research*, 18(1): 3520–3570, 2017.

Gnedenko, B. and Kolmogorov, A. *Limit Distributions for Sums of Independent Random Variables (English translation by K. L. Chung)*. Addison-Wesley, 1968.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

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

Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 1731–1741. Curran Associates, Inc., 2017.

Hu, W., Junchi Li, C., Li, L., and Liu, J.-G. On the diffusion approximation of nonconvex stochastic gradient descent. *arXiv preprint arXiv:1705.07562*, 2017.

Hu, W., Zhu, Z., Xiong, H., and Huan, J. Quasi-potential as an implicit regularizer for the loss function in the stochastic gradient descent. *arXiv preprint arXiv:1901.06054*, 2019.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. 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.

Li, Q., Tai, C., and Weinan, E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pp. 2101–2110, 2017.

Li, X., Gu, Q., Zhou, Y., Chen, T., and Banerjee, A. Hessian based analysis of sgd for deep nets: Dynamics and generalization. *arXiv preprint arXiv:1907.10732*, 2019.

Mandt, S., Hoffman, M. D., and Blei, D. M. Stochastic gradient descent as approximate bayesian inference. *The Journal of Machine Learning Research*, 18(1):4873–4907, 2017.

Martens, J. and Grosse, R. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417, 2015.

Neyshabur, B., Bhojanapalli, S., Mcallester, D., and Srebro, N. Exploring generalization in deep learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5947–5956. Curran Associates, Inc., 2017.

Nguyen, T. H., Şimşekli, U., Gürbüzbalaban, M., and Richard, G. First exit time analysis of stochastic gradient descent under heavy-tailed gradient noise. *arXiv preprint arXiv:1906.09069*, 2019.

Øksendal, B. Stochastic differential equations. In *Stochastic differential equations*, pp. 65–84. Springer, 2003.

Orvieto, A. and Lucchi, A. Continuous-time models for stochastic optimization algorithms. In *Advances in Neural Information Processing Systems*, pp. 12589–12601, 2019.

Panigrahi, A., Somani, R., Goyal, N., and Netrapalli, P. Non-gaussianity of stochastic gradient noise. *arXiv preprint arXiv:1910.09626*, 2019.

Shang, X., Zhu, Z., Leimkuhler, B., and Storkey, A. J. Covariance-controlled adaptive langevin thermostat for large-scale bayesian sampling. In *Advances in Neural Information Processing Systems*, pp. 37–45, 2015.

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

Şimşekli, U., Gürbüzbalaban, M., Nguyen, T. H., Richard, G., and Sagun, L. On the heavy-tailed theory of stochastic gradient descent for deep neural networks. *arXiv preprint arXiv:1912.00018*, 2019.

Simsekli, U., Sagun, L., and Gurbuzbalaban, M. A tail-index analysis of stochastic gradient noise in deep neural networks. *arXiv preprint arXiv:1901.06053*, 2019.

Wen, Y., Luk, K., Gazeau, M., Zhang, G., Chan, H., and Ba, J. Interplay between optimization and generalization of stochastic gradient descent with covariance noise, 2019.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Zhu, Z., Wu, J., Yu, B., Wu, L., and Ma, J. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from minima and regularization effects. *arXiv preprint arXiv:1803.00195*, 2018.

A. Missing proofs in main paper

A.1. Proof of Proposition 1

Proof. We first calculate the expectation and variance of the sampling random vector $\mathcal{W}_{\text{sgd}}$, then obtain that of the sampling noise $\mathcal{V}_{\text{sgd}}$.

Sampling with replacement In the circumstance of sampling with replacement, the sampling random vector $\mathcal{W}_{\text{sgd}}$ could be decompose as

$$\mathcal{W}_{\text{sgd}} = \mathcal{W}^1 + \cdots + \mathcal{W}^b,$$

where $\mathcal{W}^1, \ldots, \mathcal{W}^b$ are i.i.d. and each of them represents once sampling procedure. Thus $\mathcal{W}^i = (w_1^i, \ldots, w_n^i)^T$ contains one multiple of $\frac{1}{b}$ and $n-1$ multiples of zero, with random index. Hence we have

$$\mathbb{E}[w_j^i] = \frac{1}{bn}, \quad \mathbb{E}[w_j^i w_j^i] = \frac{1}{b^2 n}, \quad \mathbb{E}[w_j^i w_k^i] = 0, \ \forall j \neq k.$$

Thus

$$\begin{aligned}
\mathbb{E}[\mathcal{W}^i] =& \frac{1}{bn}\mathbb{1}, \\
\text{Var}[\mathcal{W}^i] =& \mathbb{E}[\mathcal{W}^i(\mathcal{W}^i)^T] - \mathbb{E}[\mathcal{W}^i]\mathbb{E}[\mathcal{W}^i]^T = \begin{pmatrix} \frac{1}{b^2 n} & & \\ & \ddots & \\ & & \frac{1}{b^2 n} \end{pmatrix} - \frac{1}{b^2 n^2}\mathbb{1}\mathbb{1}^T = \frac{1}{b^2 n}\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right).
\end{aligned}$$

Recall $\mathcal{W}^1, \ldots, \mathcal{W}^b$ are i.i.d., thus

$$\mathbb{E}[\mathcal{W}_{\text{sgd}}] = b\mathbb{E}[\mathcal{W}^i] = \frac{1}{n}\mathbb{1}, \quad \text{Var}[\mathcal{W}_{\text{sgd}}] = b\text{Var}[\mathcal{W}^i] = \frac{1}{bn}\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right).$$

Therefore for the sampling noise $\mathcal{V}_{\text{sgd}} = \mathcal{W}_{\text{sgd}} - \frac{1}{n}\mathbb{1}$ we have

$$\mathbb{E}[\mathcal{V}_{\text{sgd}}] = 0, \quad \text{Var}[\mathcal{V}_{\text{sgd}}] = \frac{1}{bn}\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right).$$

Sampling without replacement Let $\mathcal{W}'_{\text{sgd}} = (w'_1, \ldots, w'_n)^T$. In the case of sampling without replacement, we know the sampling random vector $\mathcal{W}'_{\text{sgd}}$ contains exactly b multiples of $\frac{1}{b}$s and $n-b$ multiples of zero, with random index. Hence we have

$$\mathbb{E}[w'_j] = \frac{\binom{n-1}{b-1}\frac{1}{b}}{\binom{n}{b}} = \frac{1}{n}, \quad \mathbb{E}[(w'_j)^2] = \frac{\binom{n-1}{b-1}\frac{1}{b^2}}{\binom{n}{b}} = \frac{1}{bn}, \quad \mathbb{E}[w'_j w'_k] = \frac{\binom{n-2}{b-2}\frac{1}{b^2}}{\binom{n}{b}} = \frac{b-1}{bn(n-1)}, \ \forall j \neq k.$$

Thus

$$\begin{aligned}
\mathbb{E}[\mathcal{W}'_{\text{sgd}}] =& \frac{1}{N}\mathbb{1}, \\
\text{Var}[\mathcal{W}'_{\text{sgd}}] =& \mathbb{E}[\mathcal{W}'_{\text{sgd}}(\mathcal{W}'_{\text{sgd}})^T] - \mathbb{E}[\mathcal{W}^i]\mathbb{E}[\mathcal{W}^i]^T \\
=& \begin{pmatrix} \frac{1}{bn} & \frac{b-1}{bn(n-1)} & \cdots & \frac{b-1}{bn(n-1)} \\ \frac{b-1}{bn(n-1)} & \frac{1}{bn} & \cdots & \frac{b-1}{bn(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{b-1}{bn(n-1)} & \frac{b-1}{bn(n-1)} & \cdots & \frac{1}{b^2 n} \end{pmatrix} - \frac{1}{n^2}\mathbb{1}\mathbb{1}^T = \frac{n-b}{bn(n-1)}\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right).
\end{aligned}$$

Therefore for the sampling noise $\mathcal{V}'_{\text{sgd}} = \mathcal{W}'_{\text{sgd}} - \frac{1}{n}\mathbb{1}$ we have

$$\mathbb{E}[\mathcal{V}'_{\text{sgd}}] = 0, \quad \text{Var}[\mathcal{V}'_{\text{sgd}}] = \frac{n-b}{bn(n-1)}\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right).$$

□

A.2. Proof of Theorem 1

Proof. Let

$$\epsilon_n = y_n - x_n^T \theta_*,$$

by assumption we have

$$\mathbb{E}[\epsilon_n x_n] = 0, \quad \mathbb{E}[\epsilon_n] = 0, \quad \mathbb{E}[\epsilon^2 x x^T] \preceq \sigma^2 \Sigma.$$

Recall the MSGD updates

$$\theta_{n+1} = \theta_n - \eta \sum_{r \in B_n} w_r \left(x_r x_r^T \theta_n - y_r x_r \right),$$

hence we have

$$\theta_{n+1} - \theta_* = \left(I - \eta \sum_{r \in B_n} w_r x_r x_r^T \right) (\theta_n - \theta_*) + \eta \sum_{r \in B_n} w_r \epsilon_r x_r.$$

Define

$$L(k) = \sum_{r \in B_k} w_r x_r x_r^T, \tag{5}$$

$$M(i,k) = \begin{cases} \left(I - \eta L(i)\right) \cdots \left(I - \eta L(k)\right), & i \geq k \\ I, & i < k. \end{cases} \tag{6}$$

$$N(k) = \sum_{r \in B_k} w_r \epsilon_r x_r, \tag{7}$$

Then recursively we obtain

$$\theta_i - \theta_* = M(i,1)(\theta_0 - \theta_*) + \eta \sum_{k=1}^{i} M(i,k+1) N(k). \tag{8}$$

Moments of $L(k)$ We first calculate the first and second moments of $N(k)$ defined in Eq. (7). Since $\mathbb{E}[w_r] = \frac{1}{B}$, $\mathbb{E}[w_i^2] = \frac{1}{bB}$, $\mathbb{E}[w_i w_j] = \frac{b-1}{bB(B-1)}$, $i \neq j$, and $\mathbb{E}\left[\|x\|_2^2 x x^T\right] \preceq R^2 \Sigma$, $\Sigma \preceq \lambda I$, we have

$$\begin{aligned}
\mathbb{E}\left[L(k)\right] =& \sum_{r=1}^{B} \mathbb{E}[w_r] \cdot \mathbb{E}[x_r x_r^T] = B \cdot \frac{1}{B} \cdot \Sigma = \Sigma. \\
\mathbb{E}\left[L(k)^2\right] =& \mathbb{E} \sum_{r=1}^{B} w_r^2 x_r x_r^T x_r x_r^T + 2\mathbb{E} \sum_{r=1}^{B-1} \sum_{s=2}^{B} w_r w_s x_r x_r^T x_s x_s^T \\
=& \sum_{r=1}^{B} \mathbb{E}[w_r^2] \cdot \mathbb{E}\left[\|x_r\|_2^2 x_r x_r^T\right] + 2 \sum_{r=1}^{B-1} \sum_{s=2}^{B} \mathbb{E}[w_r w_s] \cdot \mathbb{E}[x_r x_r^T] \cdot \mathbb{E}[x_s x_s^T] \\
=& B \cdot \frac{1}{bB} \cdot \mathbb{E}\left[\|x\|_2^2 x x^T\right] + 2 \frac{B(B-1)}{2} \cdot \frac{b-1}{bB(B-1)} \cdot \Sigma^2 \\
\preceq& \frac{1}{b} R^2 \Sigma + \frac{b-1}{b} \lambda \Sigma = \frac{R^2 + (b-1)\lambda}{b} \Sigma.
\end{aligned}$$

Moments of $M(i,k)$ We only consider $i \geq k$.

$$\begin{aligned}
\mathbb{E}[M(i,k)] =& \left(I - \eta\mathbb{E}[L(i)]\right)\cdots\left(I - \eta\mathbb{E}[L(k)]\right) = (I-\eta\Sigma)^{i-k+1}.\\
\mathbb{E}M(i,k)M(i,k)^T =& \mathbb{E}M(i,k+1)\left(I-\eta L(k)\right)^2 M(i,k+1)^T\\
=& \mathbb{E}M(i,k+1)\left(I - 2\eta L(k) + \eta^2 L(k)^2\right)M(i,k+1)^T\\
\leq& \mathbb{E}M(i,k+1)\left(I - 2\eta\Sigma + \eta^2\frac{R^2+(b-1)\lambda}{b}\Sigma\right)M(i,k+1)^T\\
=& \mathbb{E}M(i,k+1)M(i,k+1)^T - \eta\left(2-\eta\frac{R^2+(b-1)\lambda}{b}\right)\mathbb{E}M(i,k+1)\Sigma M(i,k+1)^T.
\end{aligned}$$

Hence

$$\mathbb{E}M(i,k+1)\Sigma M(i,k+1)^T \leq \frac{1}{\eta\left(2-\eta\frac{R^2+(b-1)\lambda}{b}\right)}\left(\mathbb{E}M(i,k+1)M(i,k+1)^T - \mathbb{E}M(i,k)M(i,k)^T\right).$$

Moments of $N(k)$

$$\begin{aligned}
\mathbb{E}[N(k)] =& \sum_{r\in B_k}\mathbb{E}[w_r]\cdot\mathbb{E}[\epsilon_r x_r] = B\cdot\frac{1}{B}\cdot 0 = 0.\\
\mathbb{E}\left[N(k)N(k)^T\right] =& \mathbb{E}\sum_{r=1}^{B} w_r^2\epsilon_r^2 x_r x_r^T + 2\mathbb{E}\sum_{r=1}^{B-1}\sum_{s=2}^{B} w_r w_s \epsilon_r\epsilon_s x_r x_s^T\\
=& \sum_{r=1}^{B}\mathbb{E}[w_r^2]\cdot\mathbb{E}[\epsilon_r^2 x_r x_r^T] + 2\sum_{r=1}^{B-1}\sum_{s=2}^{B}\mathbb{E}[w_r]\cdot\mathbb{E}[w_s]\cdot\mathbb{E}[\epsilon_r x_r]\cdot\mathbb{E}[\epsilon_s x_s]^T\\
=& B\cdot\frac{1}{bB}\cdot\mathbb{E}[\epsilon^2 x x^T] + 2\frac{B(B-1)}{2}\cdot\frac{1}{B^2}\cdot 0\\
\leq& \frac{1}{b}\sigma^2\Sigma.
\end{aligned}$$

Calculate averaging Takeing expectation to w_k and B_k, we have

$$\begin{aligned}
&\mathbb{E}\sum_{i=0}^{n-1}\sum_{j=i+1}^{n}\left\langle\theta_i-\theta_*, \Sigma(\theta_j-\theta_*)\right\rangle\\
=&\mathbb{E}\sum_{i=0}^{n-1}\sum_{j=i+1}^{n}\left\langle\theta_i-\theta_*, \Sigma\left(M(j,i+1)(\theta_i-\theta_*) + \eta\sum_{k=i+1}^{j}M(j,k+1)N(k)\right)\right\rangle\\
=&\mathbb{E}\sum_{i=0}^{n-1}\sum_{j=i+1}^{n}\left\langle\theta_i-\theta_*, \Sigma M(j,i+1)(\theta_i-\theta_*)\right\rangle\\
=&\mathbb{E}\sum_{i=0}^{n-1}\sum_{j=i+1}^{n}\left\langle\theta_i-\theta_*, \Sigma\left(I-\eta\Sigma\right)^{j-i}(\theta_i-\theta_*)\right\rangle\\
=&\mathbb{E}\sum_{i=0}^{n-1}\left\langle\theta_i-\theta_*, \eta^{-1}\left(I-\eta\Sigma-(I-\eta\Sigma)^{n-i+1}\right)(\theta_i-\theta_*)\right\rangle\\
\leq&\mathbb{E}\sum_{i=0}^{n}\left\langle\theta_i-\theta_*, \eta^{-1}\left(I-\eta\Sigma\right)(\theta_i-\theta_*)\right\rangle\\
=&\eta^{-1}\mathbb{E}\sum_{i=0}^{n}\|\theta_i-\theta_*\|_2^2 - \mathbb{E}\sum_{i=0}^{n}\left\|\Sigma^{\frac{1}{2}}(\theta_i-\theta_*)\right\|_2^2,
\end{aligned}$$

which implies that

$$
\begin{aligned}
&(n+1)^2\mathbb{E}\left\|\Sigma^{\frac{1}{2}}(\bar{\theta}_n-\theta_*)\right\|_2^2=\mathbb{E}\sum_{i,j=0}^{n}\left\langle\theta_i-\theta_*,\Sigma(\theta_j-\theta_*)\right\rangle\\
=&\mathbb{E}\sum_{i=0}^{n}\left\langle\theta_i-\theta_*,\Sigma(\theta_i-\theta_*)\right\rangle+2\mathbb{E}\sum_{i=0}^{n-1}\sum_{j=i+1}^{n}\left\langle\theta_i-\theta_*,\Sigma(\theta_j-\theta_*)\right\rangle\\
\leq&\mathbb{E}\sum_{i=0}^{n}\left\|\Sigma^{\frac{1}{2}}(\theta_i-\theta_*)\right\|_2^2+2\eta^{-1}\mathbb{E}\sum_{i=0}^{n}\|\theta_i-\theta_*\|_2^2-2\mathbb{E}\sum_{i=0}^{n}\left\|\Sigma^{\frac{1}{2}}(\theta_i-\theta_*)\right\|_2^2\\
\leq&2\eta^{-1}\mathbb{E}\sum_{i=0}^{n}\|\theta_i-\theta_*\|_2^2,
\end{aligned}
$$

and in the following we bound $\mathbb{E}\sum_{i=0}^{n}\|\theta_i-\theta_*\|_2^2$. We do so by bounding each term.

Now since the solution of θ_i in Eq. (8) and the fact $\mathbb{E}[N(k)]=0$, we have

$$
\begin{aligned}
\mathbb{E}\|\theta_i-\theta_*\|_2^2=&\mathbb{E}\left\|M(i,1)(\theta_0-\theta_*)\right\|_2^2+\eta^2\mathbb{E}\sum_{k=1}^{i}\sum_{j=1}^{i}\left\langle M(i,k+1)N(k),\ M(i,j+1)N(j)\right\rangle\\
=&\mathbb{E}\left\|M(i,1)(\theta_0-\theta_*)\right\|_2^2+\eta^2\mathbb{E}\sum_{k=1}^{i}\left\langle M(i,k+1)N(k),\ M(i,k+1)N(k)\right\rangle.
\end{aligned}
$$

In conclusion we have

$$
\begin{aligned}
&\frac{1}{2}\eta(n+1)^2\mathbb{E}\left\|\Sigma^{\frac{1}{2}}(\bar{\theta}_n-\theta_*)\right\|_2^2\leq\mathbb{E}\sum_{i=0}^{n}\|\theta_i-\theta_*\|_2^2\\
=&\mathbb{E}\sum_{i=0}^{n}\left\|M(i,1)(\theta_0-\theta_*)\right\|_2^2+\eta^2\mathbb{E}\sum_{i=0}^{n}\sum_{k=1}^{i}\left\langle M(i,k+1)N(k),\ M(i,k+1)N(k)\right\rangle.
\end{aligned}
$$

We call the two terms as the noiseless term and the noise term.

Noise term We bound the noise term by observing that

$$
\begin{aligned}
&\mathbb{E}\left\langle M(i,k+1)N(k),\ M(i,k+1)N(k)\right\rangle\\
=&\mathbb{E}M(i,k+1)N(k)N(k)^TM(i,k+1)^T\\
=&\mathbb{E}\mathrm{Tr}\left[N(k)N(k)^TM(i,k+1)^TM(i,k+1)\right]\\
=&\mathrm{Tr}\left[\mathbb{E}\left[N(k)N(k)^T\right]\cdot\mathbb{E}\left[M(i,k+1)^TM(i,k+1)\right]\right]\\
\preceq&\sigma^2\mathrm{Tr}\left[\Sigma\cdot\mathbb{E}\left[M(i,k+1)^TM(i,k+1)\right]\right]\\
=&\sigma^2\mathrm{Tr}\mathbb{E}\left[M(i,k+1)\Sigma M(i,k+1)^T\right]\\
\leq&\frac{\sigma^2}{\eta\left(2-\eta\frac{R^2+(b-1)\lambda}{b}\right)}\mathrm{Tr}\left(\mathbb{E}M(i,k+1)M(i,k+1)^T-\mathbb{E}M(i,k)M(i,k)^T\right).
\end{aligned}
$$

Hence

$$\begin{aligned}
&\eta^2\mathbb{E}\sum_{i=0}^{n}\sum_{k=1}^{i}\left\langle M(i,k+1)N(k),\ M(i,k+1)N(k)\right\rangle\\
\leq&\frac{\eta\sigma^2}{\left(2-\eta\frac{R^2+(b-1)\lambda}{b}\right)}\sum_{i=0}^{n}\sum_{k=1}^{i}\mathrm{Tr}\left(\mathbb{E}M(i,k+1)M(i,k+1)^T-\mathbb{E}M(i,k)M(i,k)^T\right)\\
=&\frac{\eta\sigma^2}{\left(2-\eta\frac{R^2+(b-1)\lambda}{b}\right)}\sum_{i=0}^{n}\mathrm{Tr}\left(\mathbb{E}M(i,i+1)M(i,i+1)^T-\mathbb{E}M(i,1)M(i,1)^T\right)\\
\leq&\frac{\eta\sigma^2}{\left(2-\eta\frac{R^2+(b-1)\lambda}{b}\right)}\sum_{i=0}^{n}\mathrm{Tr}\left(\mathbb{E}M(i,i+1)M(i,i+1)^T\right)\\
=&\frac{\eta\sigma^2}{\left(2-\eta\frac{R^2+(b-1)\lambda}{b}\right)}(n+1)\mathrm{Tr}[I]=\frac{\eta\sigma^2}{\left(2-\eta\frac{R^2+(b-1)\lambda}{b}\right)}(n+1)d.
\end{aligned}$$

Noiseless term Let $E_0=(\theta_0-\theta_*)(\theta_0-\theta_*)^T$. Define two linear operators S and T from symmetric matrices to symmetric matrices as

$$\begin{aligned}
SA=&\mathbb{E}\left[L(k)AL(k)\right]\\
TA=&\Sigma A+A\Sigma-\eta\mathbb{E}\left[L(k)AL(k)\right]=\Sigma A+A\Sigma-\eta SA.
\end{aligned}$$

With these notations and $M(i,1)=(I-\eta L(i))\cdots(I-\eta L(1))$, we recursively have

$$\mathbb{E}\left[M(i,1)^TM(i,1)\right]=(I-\eta T)^iI.$$

Next we bound the noiseless term

$$\begin{aligned}
\mathbb{E}\sum_{i=0}^{n}\left\|M(i,1)(\theta_0-\theta_*)\right\|_2^2=&\mathbb{E}\sum_{i=0}^{n}\mathrm{Tr}\left[M(i,1)^TM(i,1)(\theta_0-\theta_*)(\theta_0-\theta_*)^T\right]\\
=&\sum_{i=0}^{n}\left\langle\mathbb{E}M(i,1)^TM(i,1),E_0\right\rangle\\
=&\sum_{i=0}^{n}\left\langle(I-\eta T)^iI,E_0\right\rangle\\
=&\left\langle\eta^{-1}T^{-1}\left(I-(I-\eta T)^{n+1}\right)I,E_0\right\rangle\\
\leq&\left\langle\eta^{-1}T^{-1}I,E_0\right\rangle.
\end{aligned}$$

Let $M=T^{-1}I$, then $I=TM=\Sigma M+M\Sigma-\eta SM$, hence by the Kronecker's produce we have

$$I+\eta SM=\Sigma M+M\Sigma=\left(\Sigma\otimes I+I\otimes\Sigma\right)M,$$

thus

$$M=\left(\Sigma\otimes I+I\otimes\Sigma\right)^{-1}I+\left(\Sigma\otimes I+I\otimes\Sigma\right)^{-1}\eta SM=\frac{1}{2}\Sigma^{-1}+\left(\Sigma\otimes I+I\otimes\Sigma\right)^{-1}\eta SM.$$

Therefore

$$\begin{aligned}
&\mathbb{E}\sum_{i=0}^{n}\left\|M(i,1)(\theta_0-\theta_*)\right\|_2^2=\left\langle\eta^{-1}M,E_0\right\rangle=\frac{1}{2\eta}\left\langle\Sigma^{-1},E_0\right\rangle+\left\langle\left(\Sigma\otimes I+I\otimes\Sigma\right)^{-1}SM,E_0\right\rangle\\
=&\frac{1}{2\eta}(\theta_0-\theta_*)^T\Sigma^{-1}(\theta_0-\theta_*)+\left\langle SM,\left(\Sigma\otimes I+I\otimes\Sigma\right)^{-1}E_0\right\rangle.
\end{aligned}$$

We left to bound SM and $\left(\Sigma\otimes I+I\otimes\Sigma\right)^{-1}E_0$.

Bound $(\Sigma \otimes I + I \otimes \Sigma)^{-1} E_0$ By Cauchy-Schwarz inequality we have

$$E_0 = \Sigma^{\frac{1}{2}} \Sigma^{-\frac{1}{2}} (\theta_0 - \theta_*)(\theta_0 - \theta_*)^T \Sigma^{-\frac{1}{2}} \Sigma^{\frac{1}{2}} \preceq (\theta_0 - \theta_*)^T \Sigma^{-1} (\theta_0 - \theta_*) \cdot \Sigma.$$

Thus

$$(\Sigma \otimes I + I \otimes \Sigma)^{-1} E_0 \preceq (\theta_0 - \theta_*)^T \Sigma^{-1} (\theta_0 - \theta_*) \cdot (\Sigma \otimes I + I \otimes \Sigma)^{-1} \Sigma = (\theta_0 - \theta_*)^T \Sigma^{-1} (\theta_0 - \theta_*) \cdot \frac{1}{2} I.$$

Bound SM Firstly by definition,

$$\begin{aligned}
\text{Tr}[SM] =& \mathbb{E}\text{Tr}\left[L(k) M L(k)\right] = \mathbb{E} \sum_{r=1}^{B} \text{Tr}[w_r^2 x_r x_r^T M x_r x_r^T] + 2\mathbb{E} \sum_{r=1}^{B-1} \sum_{s=2}^{B} \text{Tr}[w_r w_s x_r x_r^T M x_s x_s^T] \\
=& \sum_{r=1}^{B} \mathbb{E}[w_r^2] \cdot \text{Tr}\left[\mathbb{E}\left[\|x_r\|_2^2 x_r x_r^T\right] M\right] + 2 \sum_{r=1}^{B-1} \sum_{s=2}^{B} \mathbb{E}[w_r w_s] \cdot \text{Tr}\left[\mathbb{E}[x_r x_r^T] \cdot M \cdot \mathbb{E}[x_s x_s^T]\right] \\
\leq& B \cdot \frac{1}{bB} \cdot \text{Tr}\left[R^2 \Sigma M\right] + 2 \frac{B(B-1)}{2} \cdot \frac{b-1}{bB(B-1)} \cdot \text{Tr}\left[\Sigma M \Sigma\right] \\
\leq& \frac{R^2}{b} \cdot \text{Tr}\left[\Sigma M\right] + \frac{b-1}{b} \cdot \lambda \cdot \text{Tr}\left[M\Sigma\right] = \frac{R^2 + (b-1)\lambda}{b} \text{Tr}\left[\Sigma M\right].
\end{aligned}$$

Secondly taking trace we have

$$d = \text{Tr}[I] = \text{Tr}[TM] = 2\text{Tr}[\Sigma M] - \eta \text{Tr}[SM] \geq 2\text{Tr}[\Sigma M] \geq \frac{2b}{R^2 + (b-1)\lambda} \text{Tr}[SM],$$

which implies that $\text{Tr}[SM] \leq \frac{R^2 + (b-1)\lambda}{2b} d$.

To sum up we have

$$\begin{aligned}
& \left\langle SM,\ (\Sigma \otimes I + I \otimes \Sigma)^{-1} E_0 \right\rangle \\
\leq& \frac{1}{2} (\theta_0 - \theta_*)^T \Sigma^{-1} (\theta_0 - \theta_*) \cdot \langle SM, I \rangle \\
=& \frac{1}{2} (\theta_0 - \theta_*)^T \Sigma^{-1} (\theta_0 - \theta_*) \cdot \text{Tr}[SM] \\
\leq& \frac{(R^2 + (b-1)\lambda) d}{4b} (\theta_0 - \theta_*)^T \Sigma^{-1} (\theta_0 - \theta_*).
\end{aligned}$$

Therefore for the noiseless term we have

$$\begin{aligned}
& \mathbb{E} \sum_{i=0}^{n} \left\| M(i,1)(\theta_0 - \theta_*) \right\|_2^2 = \frac{1}{2\eta} (\theta_0 - \theta_*)^T \Sigma^{-1} (\theta_0 - \theta_*) + \left\langle SM, (\Sigma \otimes I + I \otimes \Sigma)^{-1} E_0 \right\rangle \\
\leq& \left(\frac{1}{2\eta} + \frac{(R^2 + (b-1)\lambda) d}{4b} \right) (\theta_0 - \theta_*)^T \Sigma^{-1} (\theta_0 - \theta_*).
\end{aligned}$$

In conclusion we have

$$\begin{aligned}
& \frac{1}{2} \eta (n+1)^2 \mathbb{E} \left\| \Sigma^{\frac{1}{2}} (\bar{\theta}_n - \theta_*) \right\|_2^2 \leq \text{noiseless term} + \text{noise term} \\
\leq& \frac{\eta \sigma^2}{\left(2 - \eta \frac{R^2 + (b-1)\lambda}{b}\right)} (n+1) d + \left(\frac{1}{2\eta} + \frac{(R^2 + (b-1)\lambda) d}{4b} \right) (\theta_0 - \theta_*)^T \Sigma^{-1} (\theta_0 - \theta_*).
\end{aligned}$$

Hence

$$\mathbb{E} \left\| \Sigma^{\frac{1}{2}} (\bar{\theta}_n - \theta_*) \right\|_2^2 \leq \frac{1}{n+1} \cdot \frac{2\sigma^2 d}{\left(2 - \eta \frac{R^2 + (b-1)\lambda}{b}\right)} + \frac{1}{(n+1)^2} \cdot \left(1 + \frac{(R^2 + (b-1)\lambda)\eta d}{2b} \right) (\theta_0 - \theta_*)^T \Sigma^{-1} (\theta_0 - \theta_*),$$

which complete our proof.

□

B. Strong convergence of Gaussian MSGD and its SDE

Theorem 2. *(Strong convergence between Gaussian MSGD and SDE) Let $T \geq 0$. Let $C(\theta)$ be the diffusion matrix, e.g., $C(\theta) = \frac{1}{\sqrt{bN}}\nabla_\theta \mathcal{L}(\theta) \in \mathbb{R}^{D\times N}$. Assume there exist some $L, M > 0$ such that $\max_{i=1,2,...,N}(|\nabla_\theta \ell_i(\theta)|) \leq M$ and that $\nabla \ell_i(\theta)$ are Lipschitz continuous with bounded Lipschitz constant $L > 0$ uniformly for all $i = 1, 2, ..., N$.*

Then the Gaussian MSGD iteration (9)

$$\theta_{k+1} - \theta_k = -\eta \nabla_\theta L(\theta_k) + \eta C(\theta_k)\mathcal{W}_{k+1},\ \mathcal{W}_k \sim \mathcal{N}(0, I),\ i.i.d. \tag{9}$$

is a order 1 strong approximation to SDE (10)

$$\mathrm{d}\Theta_t = -\nabla_\theta L(\Theta_t)\,\mathrm{d}t + \sqrt{\eta}C(\Theta_t)\,\mathrm{d}W_t,\ \Theta_0 = \theta_0,\ W_t \in \mathbb{R}^N \text{ is a standard Brownian motion} \tag{10}$$

i.e., there exist a constant C independent on η but depending on L and M such that

$$\mathbb{E}\|\Theta_{k\eta} - \theta_k\|^2 \leq C\eta^2, \quad \text{for all } 0 \leq k \leq \lfloor T/\eta \rfloor. \tag{11}$$

Proof. We show that, as $\eta \to 0$, the discrete iteration θ_k of Eq. (9) in strong norm and on finite–time intervals is close to the solution of the SDE (10). The main techniques follow (Borkar & Mitter, 1999), but (Borkar & Mitter, 1999) only considered the case when $C(\theta)$ is a constant.

For vector $x \in \mathbb{R}^d$, we define its norm as $|x| := \sqrt{x^T x}$; for matrix $X \in \mathbb{R}^{d_1\times d_2}$, we define its norm as $|X| := \sqrt{\mathrm{Tr}(X^T X)} = \sqrt{\mathrm{Tr}(XX^T)}$.

Let $\widehat{\Theta}_t$ be the process defined by the integral form of the stochastic differential equation

$$\widehat{\Theta}_t - \widehat{\Theta}_0 = -\int_0^t \nabla_\theta L(\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor \eta})\,\mathrm{d}s + \sqrt{\eta}\int_0^t C(\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor \eta})\,\mathrm{d}W_s\ ,\ \widehat{\Theta}_0 = \theta_0\ . \tag{12}$$

Here for a real positive number $a > 0$ we define $\lfloor a \rfloor = \max\{k \in \mathbb{N}_+, k < a\}$. From (12) we see that we have, for $k = 0, 1, 2, ...$

$$\widehat{\Theta}_{(k+1)\eta} - \widehat{\Theta}_{k\eta} = -\eta\nabla_\theta L(\widehat{\Theta}_{k\eta}) - \sqrt{\eta}C(\widehat{\Theta}_{k\eta})(W_{(k+1)\eta} - W_{k\eta})\ . \tag{13}$$

Since $\sqrt{\eta}(W_{(k+1)\eta} - W_{k\eta}) \sim \mathcal{N}(0, \eta^2 I)$, we could let $\eta\mathcal{W}_{k+1} = \sqrt{\eta}(W_{(k+1)\eta} - W_{k\eta})$, where $\mathcal{W}_{k+1}$ is the i.i.d. Gaussian sequence in (9). From here, we see that

$$\widehat{\Theta}_{k\eta} = \theta_k\ , \tag{14}$$

where θ_k is the solution to (9).

We first bound $\widehat{\Theta}_t$ in Eq. (12) and Θ_t in Eq. (10). Then we could obtain the error estimation of $\theta_k = \widehat{\Theta}_{k\eta}$ and $\Theta_{k\eta}$ by simply set $t = k\eta$.

Since we assumed that $\nabla_\theta \ell_i(\theta)$ is L–Lipschitz continuous, we get $|C(\theta_1) - C(\theta_2)| = \frac{1}{\sqrt{bN}}\sqrt{\sum_{i=1}^N |\nabla_\theta \ell_i(\theta_1) - \nabla_\theta \ell_i(\theta_2)|^2} \leq \frac{1}{\sqrt{bN}}\sqrt{NL^2|\theta_1 - \theta_2|^2} \leq L|\theta_1 - \theta_2|$ since $b \geq 1$. Thus $C(\theta)$ is also L–Lipschitz continuous. Take a difference between (12) and (10) we get

$$\widehat{\Theta}_t - \Theta_t = -\int_0^t [\nabla_\theta L(\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta}) - \nabla_\theta L(\Theta_s)]\,\mathrm{d}s + \sqrt{\eta}\int_0^t [C(\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor}) - C(\Theta_s)]\,\mathrm{d}W_s\ . \tag{15}$$

We can estimate

$$\begin{aligned}
&|\nabla_\theta L(\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor \eta}) - \nabla_\theta L(\Theta_s)|^2 \\
\leq& 2|\nabla_\theta L(\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor \eta}) - \nabla_\theta L(\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta})|^2 + 2|\nabla_\theta L(\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta}) - \nabla_\theta L(\Theta_s)|^2 \\
\leq& 2L^2|\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor \eta} - \Theta_{\lfloor \frac{s}{\eta}\rfloor \eta}|^2 + 2L^2|\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta} - \Theta_s|^2\ ,
\end{aligned} \tag{16}$$

where we used the inequality $|\nabla_\theta L(\theta_1) - \nabla_\theta L(\theta_2)| \le \frac{1}{N}\sum_{i=1}^{N} |\nabla_\theta \ell_i(\theta_1) - \nabla_\theta \ell_i(\theta_2)| \le L|\theta_1 - \theta_2|$.

Similarly, we estimate

$$\begin{aligned}
&|C(\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor \eta}) - C(\Theta_s)|^2 \\
\le& 2|C(\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor \eta}) - C(\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta})|^2 + 2|C(\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta}) - C(\Theta_s)|^2 \\
\le& 2L^2|\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor \eta} - \Theta_{\lfloor \frac{s}{\eta}\rfloor \eta}|^2 + 2L^2|\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta} - \Theta_s|^2 .
\end{aligned} \tag{17}$$

On the other hand, from (15), the Itô's isometry (Øksendal, 2003) and Cauchy–Schwarz inequality we have

$$\begin{aligned}
&\mathbb{E}|\widehat{\Theta}_t - \Theta_t|^2 \\
\le& 2\mathbb{E}\left|\int_0^t [\nabla_\theta L(\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta}) - \nabla_\theta L(\Theta_s)]\,\mathrm{d}s\right|^2 + 2\eta\mathbb{E}\left|\int_0^t [C(\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor}) - C(\Theta_s)]\,\mathrm{d}W_s\right|^2 \\
\le& 2\mathbb{E}\left|\int_0^t [\nabla_\theta L(\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta}) - \nabla_\theta L(\Theta_s)]\,\mathrm{d}s\right|^2 + 2\eta\int_0^t \mathbb{E}\left|C(\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor}) - C(\Theta_s)\right|^2 \mathrm{d}s \\
\le& 2\int_0^t \mathbb{E}\left|\nabla_\theta L(\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta}) - \nabla_\theta L(\Theta_s)\right|^2 \mathrm{d}s + 2\eta\int_0^t \mathbb{E}\left|C(\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor}) - C(\Theta_s)\right|^2 \mathrm{d}s .
\end{aligned} \tag{18}$$

Combining (16), (17) and (18) we obtain that

$$\begin{aligned}
&\mathbb{E}|\widehat{\Theta}_t - \Theta_t|^2 \\
\le& 2\int_0^t \left(2L^2\mathbb{E}|\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor \eta} - \Theta_{\lfloor \frac{s}{\eta}\rfloor \eta}|^2 + 2L^2\mathbb{E}|\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta} - \Theta_s|^2\right)\mathrm{d}s \\
&+2\eta\int_0^t \left(2L^2\mathbb{E}|\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor \eta} - \Theta_{\lfloor \frac{s}{\eta}\rfloor \eta}|^2 + 2L^2\mathbb{E}|\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta} - \Theta_s|^2\right)\mathrm{d}s . \\
=& 4(1+\eta)L^2 \cdot \left(\int_0^t \mathbb{E}|\widehat{\Theta}_{\lfloor \frac{s}{\eta}\rfloor \eta} - \Theta_{\lfloor \frac{s}{\eta}\rfloor \eta}|^2\,\mathrm{d}s + \int_0^t \mathbb{E}|\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta} - \Theta_s|^2\,\mathrm{d}s\right) .
\end{aligned} \tag{19}$$

Since we assumed that there is an $M > 0$ such that $\max_{i=1,2\ldots,N}(|\nabla_\theta \ell_i(\theta)|) \le M$, we conclude that $|\nabla_\theta L(\theta)| \le \frac{1}{N}\sum_{i=1}^{N} |\nabla_\theta \ell_i(\theta)| \le M$ and $|C(\theta)| \le \frac{1}{\sqrt{bN}}\sqrt{\sum_{i=1}^{N} |\nabla_\theta \ell_i(\theta)|^2} \le M$ since $b \ge 1$. By (10), the Itô's isometry (Øksendal, 2003), the Cauchy-Schwarz inequality and $0 \le s - \lfloor \frac{s}{\eta}\rfloor \eta \le \eta$ we know that

$$\begin{aligned}
&\mathbb{E}|\Theta_{\lfloor \frac{s}{\eta}\rfloor \eta} - \Theta_s|^2 \\
=& \mathbb{E}\left|-\int_{\lfloor \frac{s}{\eta}\rfloor \eta}^{s} \nabla_\theta L(\Theta_u)\,\mathrm{d}u + \sqrt{\eta}\int_{\lfloor \frac{s}{\eta}\rfloor \eta}^{s} C(\Theta_u)\,\mathrm{d}W_u\right|^2 \\
\le& 2\mathbb{E}\left|\int_{\lfloor \frac{s}{\eta}\rfloor \eta}^{s} \nabla_\theta L(\Theta_u)\,\mathrm{d}u\right|^2 + 2\eta\mathbb{E}\left|\int_{\lfloor \frac{s}{\eta}\rfloor \eta}^{s} C(\Theta_u)\,\mathrm{d}W_u\right|^2 \\
\le& 2\mathbb{E}\left(\int_{\lfloor \frac{s}{\eta}\rfloor \eta}^{s} \left|\nabla_\theta L(\Theta_u)\right|\mathrm{d}u\right)^2 + 2\eta\int_{\lfloor \frac{s}{\eta}\rfloor \eta}^{s} \mathbb{E}|C(\Theta_u)|^2\,\mathrm{d}u \\
\le& 2\eta\int_{\lfloor \frac{s}{\eta}\rfloor \eta}^{s} \mathbb{E}|\nabla_\theta L(\Theta_u)|^2\,\mathrm{d}u + 2\eta\int_{\lfloor \frac{s}{\eta}\rfloor \eta}^{s} \mathbb{E}|C(\Theta_u)|^2\,\mathrm{d}u \\
\le& 2\eta^2 M^2 + 2\eta^2 M^2 = 4\eta^2 M^2 .
\end{aligned} \tag{20}$$

Combining (20) and (19) we obtain

$$\mathbb{E}|\widehat{\Theta}_t - \Theta_t|^2 \leq 4(1+\eta)L^2 \cdot \left(\int_0^t \mathbb{E}|\widehat{\Theta}_{\lfloor \frac{s}{\eta} \rfloor \eta} - \Theta_{\lfloor \frac{s}{\eta} \rfloor \eta}|^2 \, \mathrm{d}s + 4\eta^2 M^2 t \right) . \tag{21}$$

Set $T > 0$ and $m(t) = \max_{0 \leq s \leq t} \mathbb{E}|\widehat{\Theta}_s - \Theta_s|^2$, noticing that $m(\lfloor \frac{s}{\eta} \rfloor \eta) \leq m(s)$ (as $\lfloor \frac{s}{\eta} \rfloor \eta \leq s$), then the above gives for any $0 \leq t \leq T$,

$$m(t) \leq 4(1+\eta)L^2 \cdot \left(\int_0^t m(s) \, \mathrm{d}s + 4\eta^2 M^2 T \right) . \tag{22}$$

By Gronwall's inequality we obtain that for $0 \leq t \leq T$,

$$m(t) \leq 16(1+\eta)L^2 \eta^2 M^2 T e^{4(1+\eta)L^2 t}. \tag{23}$$

Suppose $0 < \eta < 1$, then there is a constant C which is independent on η s.t.

$$\mathbb{E}|\widehat{\Theta}_t - \Theta_t|^2 \leq m(t) \leq C\eta^2. \tag{24}$$

Set $t = k\eta$ in (24) and make use of (14), we finish the proof.

□

Remark. *As we have seen in the previous proof, the functions $\nabla_\theta L(\theta)$ and $C(\theta)$ are both L–Lipschitz continuous, and thus the SDE (10) admits a unique solution ((Øksendal, 2003), Section 5.2).*

C. Experiments setups and further results

The experiments are conducted using GeForce GTX 1080 Ti and `PyTorch 1.0.0`.

C.1. FashionMNIST

Dataset https://github.com/zalandoresearch/fashion-mnist

We randomly choose $1,000$ original test data as our training set, and use the $60,000$ original training data as our test set. Thus we have $1,000$ training data and $60,000$ test data. We scale the image data to $[0, 1]$.

Model We use a LeNet alike convolutional network:

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

Both 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 $11,330$.

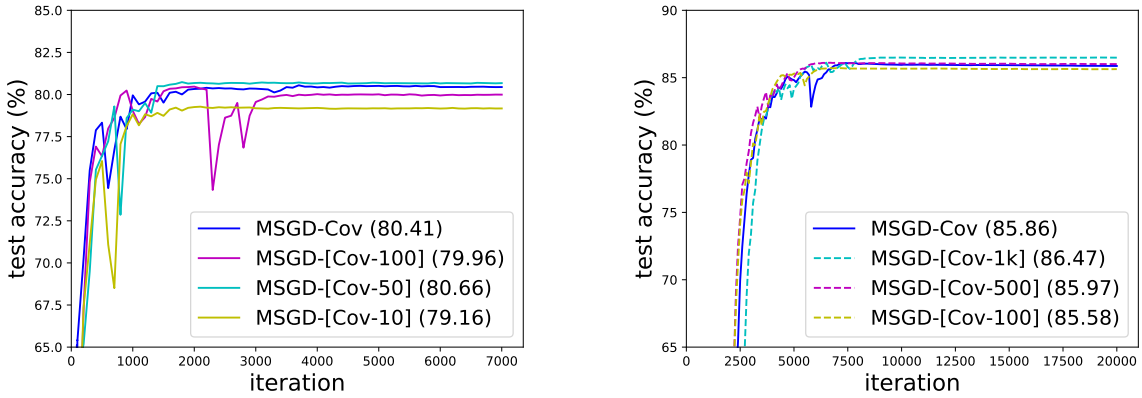
Optimization We use standard (stochastic) gradient descent optimizer. The learning rate is 0.01. If not stated otherwise, the batch size of SGD is 50.

C.2. SVHN

Dataset http://ufldl.stanford.edu/housenumbers/

We randomly choose $25,000$ original test data as our training set, and $75,000$ original training data as our test set. Thus we have $25,000$ training data and $75,000$ test data. We scale the image data to $[0, 1]$.

Model We use standard VGG-11 without Batch Normalization.



(a) Small FashionMNIST (b) SVHN

Figure 3. The generalization of MSGD. X-axis: number of iterations; y-axis: test accuracy. **(a)**: We randomly draw $1,000$ samples from FashionMNIST as the training set, then train a small convolutional network with them. **(b)**: We use $25,000$ samples from SVHN as the training set, then train a VGG-11 without Batch Normalization. **MSGD-Cov**: MSGD with Gaussian gradient noise whose covariance is the SGD covariance. **MSGD-[Cov-B]**: MSGD-Cov with the SGD covariance estimated using a mini-batch of samples in size B.

Optimization We use standard (stochastic) gradient descent optimizer. The learning rate is 0.05. If not stated otherwise, the batch size of SGD is 100.

C.3. CIFAR-10

Dataset https://www.cs.toronto.edu/~kriz/cifar.html

We use standard CIFAR-10 dataset. We scale the image into $[0, 1]$.

Models We use two models: VGG-11 without Batch Normalization and standard ResNet-18.

Optimization for VGG-11 We use momentum (stochastic) gradient descent optimizer. The momentum is 0.9. The learning rate is 0.01 decayed by 0.1 at iteration $40,000$ and $60,000$. If not stated otherwise, the batch size of SGD is 100.

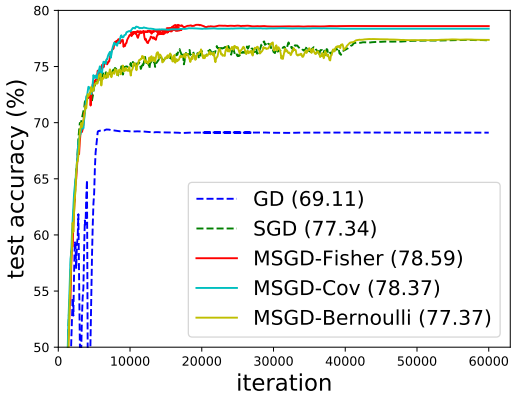
Optimization for ResNet-18 We use momentum (stochastic) gradient descent optimizer. The momentum is 0.9. The learning rate is 0.1 decayed by 0.1 at iteration $40,000$ and $60,000$. If not stated otherwise, the batch size of SGD is 100.

For large batch training, we use ghost batch normalization (Hoffer et al., 2017).

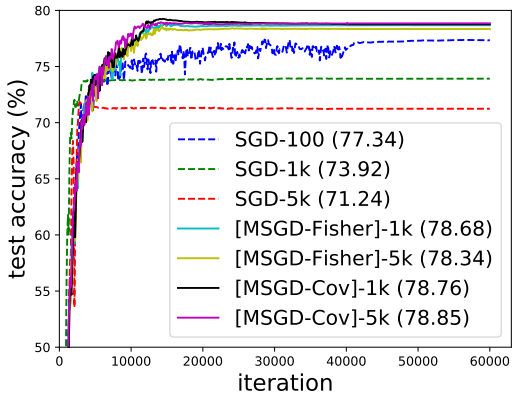
Specially, for the experiments to obtain state-of-the-art performance on ResNet-18, we also use standard data augmentation and weight decay 5×10^{-4}.

C.4. Additional experiments

Figure 3 and Figure 4 provide additional experiments. The results are consistent with our understanding in the main paper.



(a) CIFAR-10, VGG-11



(b) CIFAR-10, VGG-11

Figure 4. The generalization of MSGD and mini-batch MSGD. X-axis: number of iterations; y-axis: test accuracy. **(a) (b)**: We train a VGG-11 on CIFAR-10 without using Batch Normalization, data augmentation and weight decay. **MSGD-Fisher**: MSGD with Gaussian gradient noise whose covariance is the scaled Fisher. **MSGD-Cov**: MSGD with Gaussian gradient noise whose covariance is the SGD covariance. **MSGD-Bernoulli**: MSGD with Bernoulli sampling noise. **SGD-B**: SGD with batch size B. **[MSGD-Fisher]-B**: mini-batch MSGD with batch size B, and an compensatory gradient noise whose covariance is the estimated Fisher. textbf[MSGD-Cov]-**B**: mini-batch MSGD with batch size B, and an compensatory gradient noise whose covariance is the estimated SGD covariance.