On the SDEs and Scaling Rules for Adaptive Gradient Algorithms

Anonymous Author(s)
Affiliation
Address
email

Abstract

Approximating Stochastic Gradient Descent (SGD) as a Stochastic Differential Equation (SDE) has allowed researchers to enjoy the benefits of studying a continuous optimization trajectory while carefully preserving the stochasticity of SGD. Analogous study of adaptive gradient methods, such as RMSprop and Adam, has been challenging because there were no rigorously proven SDE approximations for these methods. This paper derives the SDE approximations for RMSprop and Adam, giving theoretical guarantees of their correctness as well as experimental validation of their applicability to common large-scaling vision and language settings. A key practical result is the derivation of a *square root scaling rule* to adjust the optimization hyperparameters of RMSprop and Adam when changing batch size, and its empirical validation in deep learning settings.

1 Introduction

Distributed synchronous optimization environments have enabled very rapid training of models (in terms of wall-clock time) by allowing a large batch size. Understanding large-batch stochastic optimization is crucial to enjoying the speed-up of this setting. In this context, [1, 2] empirically discovered the *linear scaling rule (LSR)* for Stochastic Gradient Descent (SGD). It calls for scaling learning rate proportionately to the batch size while fixing the number of epochs. It was recognized that the validity of this scaling rule stems from the benefits to generalization due to noise from mini-batch gradient estimation. But naive analysis, as done in [3], suggested that the scaling rule for SGD ought to be *square root* instead of linear. Subsequently, [4] pointed out that since the phenomenon involves varying the LR even down to zero, the correct analysis should invoke a continuous view, namely a stochastic differential equation (SDE). The SDE view helps identify the correct scaling of the noise and leads to a derivation of the linear scaling rule (see Section 2.2).

However, extending the SDE approach—i.e., continuous-time approximations—to popular adaptive optimization algorithms, like RMSprop [5] and Adam [6], has been challenging due to their use of coordinate-wise normalization. By ignoring gradient noise, [7] derived intuitive continuous approximations for full-batch RMSprop and Adam; however, this deterministic view precludes a scaling rule.

Recent papers have suggested a *square root* scaling rule for adaptive algorithms: set the learning rate proportionately to the *square root* of the batch size while fixing the number of epochs. Based on perturbation theory, [8] proposed the square root scaling rule for RMSprop and Adam but could only reason about optimization behavior near convergence, not along the entire trajectory. A square root scaling rule was also empirically discovered for another adaptive gradient method called *LAMB* [9], which is an optimization method with layerwise adaptive learning rates, designed for better optimization and generalization in large-batch training. Instead of tuning learning rates while

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

increasing batch size, LAMB used the square root scaling rule to automatically adjust the learning rate and achieve strong performance on vision and language tasks.

In this paper, we make the following contributions.

1. We propose new SDE approximations for two popular adaptive optimization algorithms, RMSprop and Adam (Definitions 4.1 and 4.3) in Theorems 4.2 and 4.4. We prove that these approximations are *1st-order weak approximations* (Definition 2.4), providing a calculus-based guarantee of the approximation strength between the SDEs and their corresponding discrete processes as was done for SGD and its SDE in [10].
2. Our SDE approximations immediately yield square-root scaling rules (Definitions 5.1 and 5.2) for adjusting the optimization hyperparameters of Adam and RMSprop when changing batch size to ensure that the resulting discrete trajectories are all 1st-order weak approximations of the same SDE (Theorem 5.3). Experiments (Figures 1 and 2 and Appendix I) validate the scaling rules in the vision and language modeling domains.
3. We provide efficient experimental verification of the validity of the SDE approximation for the adaptive algorithms in realistic deep nets (Definitions 5.1 and 5.2). Direct simulation of the SDE, e.g., Euler-Maruyama, is prohibitively expensive because it requires computing the full gradient and noise covariance at fine-grained intervals. Hence we adapt (Definition 6.2) the new and efficient *SVAG simulation* for SGD [11] for use with our proposed SDEs and rigorously prove its correctness (Theorem 6.3). Using SVAG, we provide evidence that the proposed SDE approximations track the analogous discrete trajectories in many common large-scale vision and language settings (Figure 3 and Appendix H).

2 Preliminaries

We use $\boldsymbol{v} \odot \boldsymbol{u}$, $\boldsymbol{v}^2$, $\sqrt{\boldsymbol{v}}$ to denote coordinate-wise operators for multiplication, taking squares, taking square roots. For ease of exposition we modify RMSprop and Adam to use $\boldsymbol{v}_k$ in the update for $\boldsymbol{\theta}_k$ instead of using $\boldsymbol{v}_{k+1}$.[1] We also assume that $\boldsymbol{v}_0$ is nonzero if ϵ is 0 to avoid division by zero.

Definition 2.1 (RMSprop Algorithm, [5])**.**

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \boldsymbol{g}_k \odot (\sqrt{\boldsymbol{v}_k} + \epsilon)^{-1}, \quad \boldsymbol{v}_{k+1} = \beta \boldsymbol{v}_k + (1-\beta)\boldsymbol{g}_k^2,$$

where $\boldsymbol{\theta}_k$ is the parameter, $\boldsymbol{g}_k$ is the stochastic gradient at step k, and $\boldsymbol{v}_k$ is an estimate for the second moment of $\boldsymbol{g}_k$.

Definition 2.2 (Adam Algorithm, [6])**.**

$$\boldsymbol{m}_{k+1} = \beta_1 \boldsymbol{m}_k + (1-\beta_1)\boldsymbol{g}_k, \quad \boldsymbol{v}_{k+1} = \beta_2 \boldsymbol{v}_k + (1-\beta_2)\boldsymbol{g}_k^2, \quad \widehat{\boldsymbol{m}}_{k+1} = \boldsymbol{m}_{k+1} \odot (1-\beta_1^{k+1})^{-1},$$
$$\widehat{\boldsymbol{v}}_{k+1} = \boldsymbol{v}_{k+1} \odot (1-\beta_2^{k+1})^{-1}, \quad \boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \widehat{\boldsymbol{m}}_{k+1} \odot (\sqrt{\widehat{\boldsymbol{v}}_k} + \epsilon)^{-1},$$

where $\boldsymbol{\theta}_k$ is the parameter, $\boldsymbol{g}_k$ is the stochastic gradient at step k, $\boldsymbol{m}_k$ is the momentum, and $\boldsymbol{v}_k$ is an estimate for the second moment of $\boldsymbol{g}_k$.

2.1 Noisy Gradient Oracle with Scale Parameter

We abstract the stochastic gradient as being provided by a *noisy* oracle for the full gradient. We formulate the oracle to highlight the phenomenon of interest: changing the batch size affects the scale of the noise.

Definition 2.3. A *Noisy Gradient Oracle with Scale Parameter* (NGOS) is characterized by a tuple $\mathcal{G}_\sigma = (f, \boldsymbol{\Sigma}, \mathcal{Z}_\sigma)$. Given a (noise) scale parameter $\sigma > 0$, $\mathcal{G}$ takes an input $\boldsymbol{\theta}$ and returns $\boldsymbol{g} = \nabla f(\boldsymbol{\theta}) + \sigma \boldsymbol{z}$, where $\nabla f(\boldsymbol{\theta})$ is the gradient of f at $\boldsymbol{\theta}$, $\boldsymbol{z}$ is the gradient noise drawn from the probability distribution $\mathcal{Z}_\sigma(\boldsymbol{\theta})$ with mean zero and covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. We use $\mathcal{G}_\sigma(\boldsymbol{\theta})$ to denote the distribution of $\boldsymbol{g}$ given σ and $\boldsymbol{\theta}$. The probability distribution $\mathcal{Z}_\sigma(\boldsymbol{\theta})$ can change with the scale σ, but the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is fixed across different noise scales.

For all $\boldsymbol{g}_k$ in Definitions 2.1 and 2.2, we assume that $\boldsymbol{g}_k$ is drawn from a noisy gradient oracle $\mathcal{G}_\sigma$. In our setting, as is common when batches are sampled with replacement, σ is primarily controlled

[1] Experiments in Appendix G.2 show that this change does not significantly impact performance.

through the batch size; in particular, $\sigma \sim 1/\sqrt{B}$ (see Appendix F.1 for a derivation). For sampling with replacement on a finite dataset of size n, where $f_1(\boldsymbol{\theta}), \ldots, f_n(\boldsymbol{\theta})$ are the loss functions for the n data points (and the average of these n functions is $f(\boldsymbol{\theta})$), this covariance matrix for a given parameter $\boldsymbol{\theta}$ can be explicitly written as

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(\nabla f_i(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\theta}))(\nabla f_i(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\theta}))^\top. \tag{1}$$

2.2 SDE Approximation and Scaling Rules

Under appropriate conditions it becomes possible to approximate SGD via an Itô SDE, which uses Brownian motion to model the noise and has the following general form, where W_t is a Wiener process: $\mathrm{d}\boldsymbol{X}_t = \boldsymbol{b}(\boldsymbol{X}_t)\mathrm{d}t + \boldsymbol{\sigma}(\boldsymbol{X}_t)\mathrm{d}W_t$. The SGD update rule for a loss f is $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta \boldsymbol{g}_k$, where η is the learning rate and $\boldsymbol{g}_k$ is given by the NGOS on input $\boldsymbol{x}_k$. The following is the canonical interpretation of SGD as an SDE:

$$\mathrm{d}\boldsymbol{X}_t = -\nabla f(\boldsymbol{X}_t)\mathrm{d}t + \sqrt{\eta}\boldsymbol{\Sigma}^{1/2}(\boldsymbol{X}_t)\mathrm{d}\boldsymbol{W}_t. \tag{2}$$

Equation (2) hints at a relationship between learning rate and gradient noise—specifically, the *linear scaling rule*—since scaling batch size by factor κ scales the noise covariance by $1/\kappa$, which can be canceled by scaling η by κ as well [4]. Therefore, the linear scaling rule ensures the SDE approximation does not change when using a different batch size. With the same methodology, the current paper studies the SDE approximations for adaptive gradient algorithms to derive the square root scaling rule for them.

2.3 Quality of SDE Approximation and Theoretical Assumptions

The quality of the SDE approximation can be measured empirically (Section 6) and bounded theoretically using a calculus-based guarantee, which was initiated in the context of deep learning in [10]. In particular, the theoretical guarantee uses the following notion of approximation between discrete and continuous stochastic processes by comparing iteration k in the discrete process with continuous time $k\eta_{\mathrm{e}}$, where $\eta_{\mathrm{e}} > 0$ is the (effective) step size of the discrete process.

Definition 2.4 (Order-1 Weak Approximation, [10])**.** Let $\{\boldsymbol{X}_t^{\eta_{\mathrm{e}}} : t \in [0, T]\}$ and $\{\boldsymbol{x}_k^{\eta_{\mathrm{e}}}\}_{k=0}^{\lfloor T/\eta_{\mathrm{e}}\rfloor}$ be families of continuous and discrete stochastic processes parametrized by η_{e}. We say $\{\boldsymbol{X}_t^{\eta_{\mathrm{e}}}\}$ and $\{\boldsymbol{x}_k^{\eta_{\mathrm{e}}}\}$ are order-1 weak approximations of each other if for every test function g with at most polynomial growth (Definition B.1), there exists a constant $C > 0$ independent of η_{e} such that

$$\max_{k=0,\ldots,\lfloor T/\eta_{\mathrm{e}}\rfloor} |\mathbb{E}g(\boldsymbol{x}_k^{\eta_{\mathrm{e}}}) - \mathbb{E}g(\boldsymbol{X}_{k\eta_{\mathrm{e}}}^{\eta_{\mathrm{e}}})| \leq C\eta_{\mathrm{e}}$$

A function $g\colon \mathbb{R}^d \to \mathbb{R}$ is said to have *polynomial growth* if there exist positive integers $\kappa_1, \kappa_2 > 0$ such that $|g(\boldsymbol{x})| \leq \kappa_1(1 + \|\boldsymbol{x}\|_2^{2\kappa_2})$ for all $\boldsymbol{x} \in \mathbb{R}^d$ (Definition B.1). The above definition measures the strength of the approximation by the closeness of a test function g computed on the iterates of both trajectories. The approximation becomes stronger in this sense as η_{e} becomes smaller. In the SDE approximation of SGD, $\eta_{\mathrm{e}} = \eta$ and k steps correspond to continuous time $k\eta$. A key difference between SGD and adaptive algorithms is that $\eta_{\mathrm{e}} = \eta^2$ for both RMSprop and Adam, which means k steps correspond to continuous time $k\eta^2$. We validate this time-scaling theoretically in Section 4.

Now we formalize the assumptions needed in the theory. Since our analysis framework is based upon calculus, it becomes necessary to assume differentiability conditions on the NGOS (Definition 2.5). Similar differentiability conditions also appear in prior SDE works [10, 11], and we note that lately it has become clear that restricting to differentiable losses (via differentiable node activations such as Swish [12]) does not hinder good performance.

Definition 2.5 (Well-behaved NGOS)**.** The loss function f and covariance matrix function $\boldsymbol{\Sigma}$ in a NGOS $\mathcal{G}_\sigma$ are *well-behaved* if they satisfy[2]: (1) $\nabla f(\boldsymbol{\theta})$ is Lipschitz and $\mathcal{C}^\infty$-smooth; (2) The square root of covariance matrix $\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\theta})$ is bounded, Lipschitz, and $\mathcal{C}^\infty$-smooth; and (3) All partial derivatives of $\nabla f(\boldsymbol{\theta})$ and $\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\theta})$ up to and including the 4-th order have polynomial growth. We also say that the NGOS $\mathcal{G}_\sigma$ is well-behaved if f and $\boldsymbol{\Sigma}$ are well-behaved.

[2]Note: $\mathcal{C}^\infty$-smoothness can be relaxed using the mollification technique from [10].

Deriving an SDE approximation also requires conditions on the noise distribution in the NGOS. It is allowed to be non-Gaussian, but not heavy-tailed. We require an upper bound on the third moment of the noise so that the distribution is not very skewed. For other higher order moments, we require $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{Z}_\sigma}[\|\boldsymbol{z}\|_2^p]^{1/p}$, namely the L^p-norm of random variable $\|\boldsymbol{z}\|_2$, to grow at most linearly as a function of $\boldsymbol{\theta}$ (which is implied by ensuring an upper bound on all even order moments). We note that the following conditions are standard in prior work using Itô SDEs to study SGD.

Definition 2.6 (Low Skewness Condition)**.** The NGOS $\mathcal{G}_\sigma$ is said to satisfy the *low skewness condition* if there is a function $K_3(\boldsymbol{\theta})$ of polynomial growth (independent of σ) such that $|\mathbb{E}_{\boldsymbol{z}\sim\mathcal{Z}_\sigma(\boldsymbol{\theta})}[\boldsymbol{z}^{\otimes 3}]| \leq K_3(\boldsymbol{\theta})/\sigma$ for all $\boldsymbol{\theta} \in \mathbb{R}^d$ and all noise scale parameters σ.

Definition 2.7 (Bounded Moments Condition)**.** The NGOS $\mathcal{G}_\sigma$ is said to satisfy the *bounded moments condition* if for all integers $m \geq 1$ and all noise scale parameters σ, there exists a constant C_{2m} (independent of σ) such that $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{Z}_\sigma(\boldsymbol{\theta})}[\|\boldsymbol{z}\|_2^{2m}]^{\frac{1}{2m}} \leq C_{2m}(1 + \|\boldsymbol{\theta}\|_2)$ for all $\boldsymbol{\theta} \in \mathbb{R}^d$.

For well-behaved loss $f(\boldsymbol{\theta})$ and covariance $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, the above two conditions are satisfied when $\mathcal{Z}_\sigma$ is the Gaussian distribution with mean zero and covariance $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. That is, the Gaussian NGOS $\boldsymbol{g} \sim \mathcal{N}(\nabla f(\boldsymbol{\theta}), \sigma^2\boldsymbol{\Sigma}(\boldsymbol{\theta}))$ satisfies the low skewness and bounded moments conditions. The low skewness condition holds because the odd moments of a Gaussian are all zeros, and the bounded moments condition can be verified since $\mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{\Sigma}(\boldsymbol{\theta}))}[\|\boldsymbol{z}\|_2^{2m}]^{\frac{1}{2m}} \leq \mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}[\|\boldsymbol{w}\|_2^{2m}]^{\frac{1}{2m}} \cdot \|\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\theta})\|_2$ and $\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\theta})$ is Lipschitz.

The Gaussian NGOS with $\sigma = \frac{1}{\sqrt{B}}$ can be seen as an approximation of the gradient noise in a mini-batch training with batch size B, if $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is set to match with (1). But this does not directly imply that the gradient noise in mini-batch training satisfies the low skewness and bounded moments conditions, as the noise is not exactly Gaussian. Assuming that the gradient of the loss function $f_i(\boldsymbol{\theta})$ at every data point is Lipschitz, these two conditions can indeed be verified for all batch sizes $B \geq 1$.

One limitation of our assumptions is that $K_3(\boldsymbol{\theta})$ and C_{2m} could be large in practice, meaning higher order moments of the gradient noise have non-negligible effects on training. [13] and [14] presented experimental evidence that the noise is heavy-tailed[3]. However, [11, 16] highlighted issues with the evidence, noting that the measurements used in [14] are intended only for scalar values. When applied to vector valued distributions the measurement can (incorrectly) identify a multi-dimensional Gaussian distribution as heavy-tailed too [11]. Hence, the heavy-tailedness of real-world gradient noise remains an open question. The success of our empirical validations of the SDE approximations, through the scaling rule and SVAG, suggests that our assumptions are not too strong. We leave it for future work to investigate if and how heavy-tailed noise plays a role in adaptive optimization.

3 Related Work

We defer a full discussion of empirical and theoretical works on adaptive gradient methods to Appendix A.1 and only discuss works relevant to SDEs and scaling rules here.

Applications of SDE Approximations. There are applications of the SDE approximation beyond the derivation of a scaling rule. [17] and [18] assumed that the loss has some symmetry (i.e., scale invariance) and studied the resulting dynamics. Furthermore, [17] used this property to explain the phenomenon of sudden rising error after LR decay in training. [16] analyzed why SGD favors flat minima using an SDE-motivated diffusion model.

Past Work on Square Root Scaling Rule. As mentioned before, square root scaling was incorrectly believed for a few years to be theoretically justified for SGD. [8] decomposed the stochastic Hessian during batch training into a deterministic Hessian and stochastic sampling perturbation and assumes one of the components to be low rank to propose a square root scaling rule. [9] empirically discovered a square root scaling rule for language models trained by LAMB, a layer-wise variant of Adam. [19] heuristically derived, but did not show an approximation bound for, a second-order SDE for

[3]This motivated [15] to consider a Lévy SDE approximation (instead of Itô SDE) to study the (worse) generalization behavior of Adam. However, the quality of the Lévy SDE approximation was not formally guaranteed (e.g., in the sense of Definition 2.4), so finding a guaranteed approximation for adaptive optimization algorithms remains an open problem.

approximating Adam, and they applied the SDE to study the time needed for Adam to escape sharp minima. [19] did not discuss a scaling rule, though their proposed SDE may admit one. Similarly, [15] derived a Lévy SDE for Adam, but no approximation bounds are given in the paper.

4 SDEs for Adaptive Algorithms

An SDE approximation operates in continuous time and thus implicitly considers the limit $\eta \to 0$. In adaptive algorithms, the moment averaging parameters β, β_1, β_2 and η must be taken to limits such that the adaptivity and stochasticity can still be studied. For example, if β, β_1, β_2 remain fixed when $\eta \to 0$, then the algorithm computes the moving averages in a very small neighborhood, which averages out the effects of gradient noise and gradient history, causing the flow to turn into deterministic SignGD [7]. We will need to assume $\beta, \beta_1, \beta_2 \to 1$ as $\eta \to 0$, which implies the impact of the history grows as the learning rate decreases, and thus the adaptive features of these algorithms can still be studied in the continuous approximation [7]. To keep the stochastic nature of the flow, we require the noise from mini-batching dominate the gradient updates.

4.1 Warm-Up: Linear loss

To build intuition for the SDE and the scaling rule, we first study a simplified setting. In particular, consider a linear loss $f(\boldsymbol{\theta}) = \langle \boldsymbol{\theta}, \bar{\boldsymbol{g}} \rangle$ and isotropic noise in the NGOS, namely $\boldsymbol{g}_k \sim \mathcal{N}(\bar{\boldsymbol{g}}, \sigma^2 \boldsymbol{I})$. The RMSprop update $\boldsymbol{v}_{k+1} = \beta \boldsymbol{v}_k + (1-\beta)\boldsymbol{g}_k^2$ can be expanded as $\boldsymbol{v}_k = \beta^k \boldsymbol{v}_0 + (1-\beta)\sum_{j=0}^{k-1} \beta^j \boldsymbol{g}_j^2$. By linearity of expectation,

$$\mathbb{E}[\boldsymbol{v}_k] = \beta^k \boldsymbol{v}_0 + (1-\beta)\sum_{j=0}^{k-1} \beta^j (\bar{\boldsymbol{g}}^2 + \sigma^2 \mathbf{1}) = \beta^k \boldsymbol{v}_0 + (1-\beta^k)(\bar{\boldsymbol{g}}^2 + \sigma^2 \mathbf{1}).$$

This suggests that $\mathbb{E}[\boldsymbol{v}_k]$ is approximately $\bar{\boldsymbol{g}}^2 + \sigma^2 \mathbf{1}$ after a sufficient number of steps. Setting $\boldsymbol{v}_0 = \bar{\boldsymbol{g}}^2 + \sigma^2 \mathbf{1}$, we see that the approximation $\mathbb{E}[\boldsymbol{v}_k] = \bar{\boldsymbol{g}}^2 + \sigma^2 \mathbf{1}$ becomes exact for all $k \geq 0$.

Using the linearity of variance (for independent variables), the standard deviation of each coordinate of $\boldsymbol{v}_k$ is of scale $\mathcal{O}((1-\beta)\sigma^2)$. Moreover, the expectation $\mathbb{E}[\boldsymbol{v}_k]$ is of scale $\mathcal{O}(\sigma^2)$, so we know that $\boldsymbol{v}_k$ is nearly deterministic and concentrates around its expectation $\bar{\boldsymbol{g}}^2 + \sigma^2 \mathbf{1}$ when β is close to 1. Therefore, we take the approximation $\boldsymbol{v}_k \approx \bar{\boldsymbol{g}}^2 + \sigma^2 \mathbf{1}$ for all $k \geq 0$. Ignoring ϵ, the RMSprop update rule becomes:

$$\boldsymbol{\theta}_{k+1} \approx \boldsymbol{\theta}_k - \eta \boldsymbol{g}_k \odot (\bar{\boldsymbol{g}}^2 + \sigma^2 \mathbf{1})^{-1/2}. \tag{3}$$

These dynamics depend on the relative magnitudes of $\bar{\boldsymbol{g}}$ and σ. We show that when $\sigma \ll \|\bar{\boldsymbol{g}}\|$, no SDE approximation can exist in Appendix F.2. Here, we explore the case where $\sigma \gg \|\bar{\boldsymbol{g}}\|$ which implies $\boldsymbol{\theta}_{k+1} \approx \boldsymbol{\theta}_k - \frac{\eta}{\sigma}\boldsymbol{g}_k$. Noting that $\boldsymbol{g}_k \sim \mathcal{N}(\bar{\boldsymbol{g}}, \sigma^2 \boldsymbol{I})$ gives $\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k \sim \mathcal{N}(\frac{\eta}{\sigma}\bar{\boldsymbol{g}}, \eta^2 \boldsymbol{I})$ approximately. Since Gaussian variables are additive, we can take a telescoping sum to obtain the marginal distribution of $\boldsymbol{\theta}_k$: $\boldsymbol{\theta}_k \sim \mathcal{N}\left((k\eta/\sigma)\bar{\boldsymbol{g}}, k\eta^2 \boldsymbol{I}\right)$ approximately.

If an SDE approximation of RMSprop exists, then $\boldsymbol{\theta}_k$ can be closely approximated by a *fixed* random variable from the corresponding stochastic process at a fixed (continuous) time t. Thus, to keep the SDE unchanged upon changing the noise scale σ, the hyperparameters must be adjusted to keep $\frac{k\eta}{\sigma}$ and $k\eta^2$ unchanged, which implies $\eta \sim \frac{1}{\sigma}$ and $k \sim \frac{1}{\eta^2}$. These observations intuitively yield the square root scaling rule: noting that σ changes with mini-batch size B as $\sigma \sim 1/\sqrt{B}$ suggests $\eta \sim \sqrt{B}$, and $k \sim 1/B$.

4.2 SDE Approximations for Adaptive Algorithms

Having established intuition in a highly simplified setting for adaptive algorithms, we now return to a more general and realistic setting. We derive the SDEs that are order-1 approximations of the discrete RMSprop and Adam algorithms under Definition 2.5, where the SDE states consist of both the parameters $\boldsymbol{\theta}$ and moment estimates. From the example of Section 4.1, we see that SDE approximations may exist if $\sigma \sim 1/\eta^2$ and $\sigma \gg \|\bar{\boldsymbol{g}}\|$ (see Appendix G.1 for empirical validation of this assumption). In this case, we can prove that $k \sim \eta^2$ is true not only for the simple setting above but also in general. This is a key difference to the SDE for SGD — each step in RMSprop or Adam

corresponds to a time interval of η^2 in SDEs, but each SGD step corresponds to a time interval of η. In Section 4.1, $\boldsymbol{v} \sim \sigma^2 \sim 1/\eta^2$ grows to infinity as $\eta \to 0$. This also happens in the general setting, so we track $\boldsymbol{u}_k \triangleq \boldsymbol{v}_k/\sigma^2$ (instead of $\boldsymbol{v}_k$ directly) in the SDEs.

Definition 4.1 (SDE for RMSprop)**.** Let $\sigma_0 \triangleq \sigma\eta$, $\epsilon_0 \triangleq \epsilon\eta$, and $c_2 \triangleq (1-\beta)/\eta^2$. Define the state of the SDE as $\boldsymbol{X}_t = (\boldsymbol{\theta}_t, \boldsymbol{u}_t)$ and the dynamics as

$$\mathrm{d}\boldsymbol{\theta}_t = -\boldsymbol{P}_t^{-1}\left(\nabla f(\boldsymbol{\theta}_t)\mathrm{d}t + \sigma_0\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\theta}_t)\mathrm{d}\boldsymbol{W}_t\right), \qquad \mathrm{d}\boldsymbol{u}_t = c_2(\mathrm{diag}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t)) - \boldsymbol{u}_t)\mathrm{d}t$$

where $\boldsymbol{P}_t := \sigma_0\mathrm{diag}(\boldsymbol{u}_t)^{1/2} + \epsilon_0\boldsymbol{I}$ is a preconditioner matrix, and $\boldsymbol{W}_t$ is the Wiener process.

Theorem 4.2 (Informal version of Theorem C.1)**.** *Let $\boldsymbol{u}_k \triangleq \boldsymbol{v}_k/\sigma^2$ and define the state of the discrete RMSprop trajectory with hyperparameters η, β, ϵ (Definition 2.1) as $\boldsymbol{x}_k = (\boldsymbol{\theta}_k, \boldsymbol{u}_k)$. Then, for a well-behaved NGOS (Definition 2.3) satisfying the skewness and bounded moments conditions (Definitions 2.6 and 2.7), the SDE in Definition 4.1 satisfies*

$$\max_{k=0,\ldots,\lfloor T/\eta^2\rfloor} |\mathbb{E}g(\boldsymbol{x}_k) - \mathbb{E}g(\boldsymbol{X}_{k\eta^2})| \le C\eta^2$$

where g and T are defined as in Definition 2.4.

Proof. We provide a proof sketch here and defer the technical details to Theorem C.1. The proof follows the same steps as [10]: first, we compute the approximation error of the continuous trajectory after one step in discrete time. Then, we use the single step error to bound the error over a finite interval of time. The proof additionally relies on constructing an auxiliary SDE with an equivalent distribution to the desired SDE (Theorem C.3) but with better smoothness conditions, and we prove this SDE to be an order-1 weak approximation of the discrete trajectory. □

We need to find continuous approximations of the normalization constants in Adam (Definition 2.2). As in the RMSprop case, we take $(1-\beta_2)/\eta^2 = c_2$. Then, we can estimate the normalization constant $1-\beta_2^k$ in continuous time $t = k\eta^2$ as $1-\beta_2^k = 1-(1-c_2\eta^2)^{t/\eta^2} \approx 1-\exp(-c_2 t)$. We can do the analogous approximation for the other normalization constant $1-\beta_1^k$ in Adam. Taking $(1-\beta_1)/\eta^2 = c_1$, we can approximate it as $1-\beta_1^k \approx 1-\exp(-c_1 t)$. This is a heuristic approach to deal with the normalization constants, but we can indeed justify it in theory.

Definition 4.3 (Adam SDE)**.** Let $c_1 \triangleq (1-\beta_1)/\eta^2$ and define $\boldsymbol{u}_k, \sigma_0, c_2, \epsilon_0, \boldsymbol{\Sigma}(\boldsymbol{\theta}_t)$ as in Definition 4.1. Let $\gamma_1(t) \triangleq 1-\exp(-c_1 t)$ and $\gamma_2(t) \triangleq 1-\exp(-c_2 t)$. Define the state of the SDE as $\boldsymbol{X}_t = (\boldsymbol{\theta}_t, \boldsymbol{m}_t, \boldsymbol{u}_t)$ and the dynamics as

$$\mathrm{d}\boldsymbol{\theta}_t = -\frac{\sqrt{\gamma_2(t)}}{\gamma_1(t)}\boldsymbol{P}_t^{-1}\boldsymbol{m}_t\mathrm{d}t, \qquad \mathrm{d}\boldsymbol{m}_t = c_1(\nabla f(\boldsymbol{\theta}_t) - \boldsymbol{m}_t)\mathrm{d}t + \sigma_0 c_1\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\theta}_t)\mathrm{d}W_t,$$
$$\mathrm{d}\boldsymbol{u}_t = c_2(\mathrm{diag}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t)) - \boldsymbol{u}_t)\mathrm{d}t,$$

where $\boldsymbol{P}_t := \sigma_0\mathrm{diag}(\boldsymbol{u}_t)^{1/2} + \epsilon_0\sqrt{\gamma_2(t)}\boldsymbol{I}$ is a preconditioner matrix, $\boldsymbol{W}_t$ is the Wiener process.

Our main theorem for Adam is given below. The initial steps of the discrete Adam trajectory can be discontinuous and noisy because of the normalization constants changing drastically. Hence, we introduce a constant t_0 and show that for any t_0, we can construct an SDE to be a weak approximation for Adam starting from the $\lceil t_0/\eta^2\rceil$-th step of Adam.

Theorem 4.4 (Informal version of Theorem D.1)**.** *Define $\boldsymbol{u}_k = \boldsymbol{v}_k/\sigma^2$ and let $\boldsymbol{x}_k = (\boldsymbol{\theta}_k, \boldsymbol{m}_k, \boldsymbol{u}_k) \in \mathbb{R}^{3d}$ be the state of the discrete Adam trajectory with hyperparameters $\eta, \beta_1, \beta_2, \epsilon$. Then, for a well-behaved NGOS (Definition 2.3) satisfying the skewness and bounded moments conditions (Definitions 2.6 and 2.7) and any $t_0 > 0$, the SDE in Definition 4.3 satisfies*

$$\max_{k=\lceil t_0/\eta^2\rceil,\ldots,\lfloor T/\eta^2\rfloor} |\mathbb{E}g(\boldsymbol{x}_k) - \mathbb{E}g(\boldsymbol{X}_{k\eta^2})| \le C\eta^2$$

where g and T are defined as in Definition 2.4 and the initial condition of the SDE is $\boldsymbol{X}_{t_0} = x_{\lceil t_0/\eta^2\rceil}$.

5 Square Root Scaling Rule

The SDE approximations in Definitions 4.1 and 4.3 motivate scaling rules for how to adjust the optimization hyperparameters when changing the batch size. In order for σ_0, c_1, c_2, and ϵ_0 to remain constant, as required by the SDEs, one needs to change $\eta, \beta, \beta_1, \beta_2$, and ϵ accordingly.

Definition 5.1 (RMSprop Scaling Rule)**.** When running RMSprop (Definition 2.1) with batch size $B' = \kappa B$, use the hyperparameters $\eta' = \eta\sqrt{\kappa}$, $\beta' = 1 - \kappa(1-\beta)$, and $\epsilon' = \frac{\epsilon}{\sqrt{\kappa}}$.

Definition 5.2 (Adam Scaling Rule)**.** When running Adam (Definition 2.2) with batch size $B' = \kappa B$, use the hyperparameters $\eta' = \eta\sqrt{\kappa}$, $\beta_1' = 1 - \kappa(1-\beta_1)$, $\beta_2' = 1 - \kappa(1-\beta_2)$, and $\epsilon' = \frac{\epsilon}{\sqrt{\kappa}}$.

Theorem 5.3 (Validity of the Scaling Rules)**.** *Suppose we have a well-behaved NGOS satisfying the low skewness and bounded moments conditions.*

1. *Let $\boldsymbol{x}_k^{(B)}$ be the discrete RMSprop (Definition 2.1) trajectory with batch size B and hyperparameters η, β, and ϵ. Let $\boldsymbol{x}_k^{(\kappa B)}$ be the trajectory with batch size κB and hyperparameters adjusted according to Definition 5.1. If $\boldsymbol{x}_k^{(B)}$ and $\boldsymbol{x}_k^{(\kappa B)}$ start from the same initial point, then with g and T defined as in Definition 2.4, $\max_{k=0,\ldots,\lfloor T/\eta^2 \rfloor} \left| \mathbb{E}g(\boldsymbol{x}_k^{(B)}) - \mathbb{E}g(\boldsymbol{x}_{\lfloor k/\kappa \rfloor}^{(\kappa B)}) \right| \leq C(1+\kappa)\eta^2$.*

2. *Fix $t_0 > 0$. Let $\boldsymbol{x}_k^{(B)}$ be the discrete Adam (Definition 2.2) trajectory with batch size B and hyperparameters η, β_1, β_2, and ϵ. Let $\boldsymbol{x}_k^{(\kappa B)}$ be the trajectory with batch size κB and hyperparameters adjusted according to Definition 5.2. If $\boldsymbol{x}_{\lceil t_0/\eta^2 \rceil}^{(B)}$ and $\boldsymbol{x}_{\lceil \kappa t_0/\eta^2 \rceil}^{(\kappa B)}$ are equal, then with g and T defined as in Definition 2.4, $\max_{k=\lceil t_0/\eta^2 \rceil,\ldots,\lfloor T/\eta^2 \rfloor} |\mathbb{E}g(\boldsymbol{x}_k^{(B)}) - \mathbb{E}g(\boldsymbol{x}_{\lfloor k/\kappa \rfloor}^{(\kappa B)})| \leq C(1+\kappa)\eta^2$.*

Proof. By the linearity of covariance, scaling the batch size by κ only modifies the NGOS by scaling σ by $1/\sqrt{\kappa}$. Hence, both scaling rules ensure that σ_0, c_1, c_2, and ϵ_0 (and thus, the SDEs) are unchanged when the batch size changes. The approximation bounds in Theorems 4.2 and 4.4 are in terms of η^2, and since η is scaled here by $\sqrt{\kappa}$, the same method gives an upper bound $C\kappa\eta^2$. Adding the approximation bounds for η and $\sqrt{\kappa}\eta$ together gives $C(1+\kappa)\eta^2$. □

Remark 5.4. The t_0 condition on the Adam rule, a holdover from the condition in Theorem 4.4, implies that our theory only directly applies when there is a warm-start phase of $\lceil t_0/\eta^2 \rceil$, where the marginal distribution of the trainable parameters at the end of the phase is the same across different learning rates η. Regardless, the scaling rules are shown to work in practice even without this phase.

The scaling rules depend on maintaining the same SDE approximation, so the bounded moments and low skewness conditions are sufficient (but not necessary) for this scaling rule to work. [11] provided an analogous discussion for SGD, and they show the scaling rule can hold even if there is heavy-tailed noise. We leave a study of heavy-tailed gradient noise in adaptive algorithms as future work.

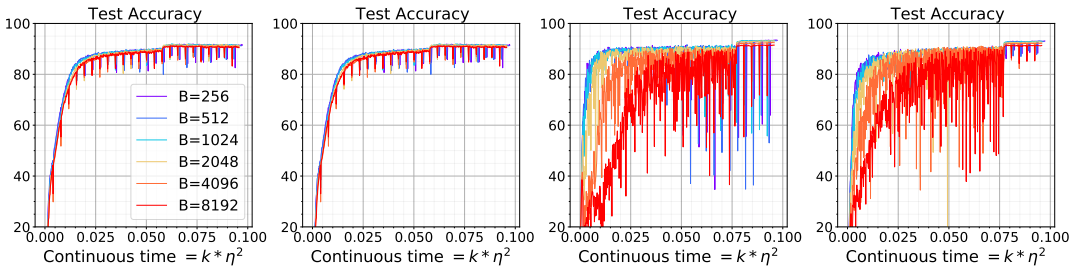
Experiments. Figures 1 and 2 show the square root scaling rule applied to ResNet-50 [20] and VGG-16 [21] trained on CIFAR-10 [22], RoBERTa-large [23] trained on the Wiki+Books corpus [24], 12-layer GPT [25] on WikiText-103 [26] and ResNet-50 trained on ImageNet [27]. We use the efficient language model pre-training recipe outlined in [28]. Appendix I has many additional settings, including ablations against other scaling rules (Appendix I.1).

6 SVAG for Adaptive Algorithms

Validating the approximation strength captured in Definition 2.4 involves comparing the discrete algorithms and their SDEs on a set of test functions. However, obtaining the SDE solution through traditional simulations, e.g., Euler-Maruyama, is computationally intractable.[4]

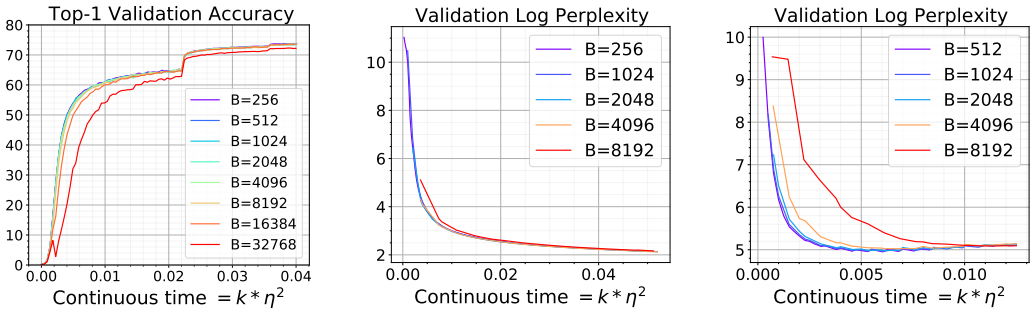
[11] proposed an efficient simulation, SVAG, of the SDE for SGD in the finite LR regime: scale the constant LR by $1/\ell$ and take the limit $\ell \to \infty$. In practice the simulation converges for a small

[4] One can also simulate the SDE by constructing 1st-order weak approximations while taking $\eta \to 0$ along the scaling rules, but the batch size cannot be smaller than 1 and hence η cannot go arbitrarily close to the limit.



(a) VGG-16 with Adam (b) ResNet-50 with Adam (c) VGG-16 with RMSprop (d) ResNet-50 with RMSprop

Figure 1: Square root scaling rule experiments on CIFAR-10 with VGG-16 and ResNet-50 (details in Appendix J). Same color legend has been used across all the plots. The performance gap between $B = 256$ and $B = 8192$ is at most 3% in all cases.



(a) ResNet-50 on ImageNet (b) RoBERTa on Wikipedia + Bookcorpus (c) GPT on WikiText-103

Figure 2: Large scale square root scaling rule experiments (details in Appendix J). Small and large batch models differ by at most 1.5% test accuracy in vision and 0.5 perplexity in language.

value of ℓ. We adapt SVAG technique to simulate our proposed SDEs, which requires additionally adjusting the moment averaging hyperparameters (i.e., β, β_1, β_2) and ϵ.

Definition 6.1 (SVAG Operator)**.** Given an NGOS $\mathcal{G}_\sigma = (f, \boldsymbol{\Sigma}, \mathcal{Z}_\sigma)$ with scale σ (Definition 2.3) and hyperparameter $\ell \geq 1$, the SVAG operator transforms $\mathcal{G}_\sigma$ into an NGOS $\widehat{\mathcal{G}}_{\ell\sigma} = (f, \boldsymbol{\Sigma}, \widehat{\mathcal{Z}}_{\ell\sigma})$ with scale $\ell\sigma$. The NGOS $\widehat{\mathcal{G}}_{\ell\sigma}$ takes an input $\boldsymbol{\theta}$ and returns $\hat{\boldsymbol{g}} = r_1(\ell)\boldsymbol{g}_1 + r_2(\ell)\boldsymbol{g}_2$, where $\boldsymbol{g}_1, \boldsymbol{g}_2$ are two stochastic gradients from $\mathcal{G}_\sigma(\boldsymbol{\theta})$ and $r_i(\ell) = \frac{1}{2}(1 + (-1)^i\sqrt{2\ell^2 - 1})$ for $i \in \{1, 2\}$. The probability distribution $\widehat{\mathcal{Z}}_{\ell\sigma}$ is defined such that $\hat{\boldsymbol{g}}$ has the same distribution as $\nabla f(\boldsymbol{\theta}) + \ell\sigma\boldsymbol{z}$ when $\boldsymbol{z} \sim \widehat{\mathcal{Z}}_{\ell\sigma}(\boldsymbol{\theta})$.

Lemma E.1 verifies that $\widehat{G}_{\ell\sigma}$ does indeed compute stochastic gradients for f with covariance $\boldsymbol{\Sigma}$. Applying the SVAG operator to mini-batch training amplifies the noise scale by ℓ. We then apply the square root scaling rule to adjust the learning rates and other hyperparameters accordingly, which yields the SVAG algorithm.

Definition 6.2 (SVAG Algorithm)**.** For a loss f, SVAG operator hyperparameter $\ell > 0$, and optimization hyperparameters $\eta, \beta, \beta_1, \beta_2$, and ϵ, compute the stochastic gradient as $\hat{\boldsymbol{g}} = r_1(\ell)\boldsymbol{g}_{\gamma_1} + r_2(\ell)\boldsymbol{g}_{\gamma_2}$, where r_1 and r_2 are defined as in Definition 6.1, and scale the optimization hyperparameters:

1. For RMSprop, set $\eta \leftarrow \eta/\ell$, $\beta \leftarrow 1-(1-\beta)/\ell^2$, and $\epsilon \leftarrow \epsilon\ell$ and apply updates as in Definition 2.1.
2. For Adam, set $\eta \leftarrow \eta/\ell$, $\beta_1 \leftarrow 1 - (1 - \beta_1)/\ell^2$, $\beta_2 \leftarrow 1 - (1 - \beta_2)/\ell^2$ and $\epsilon \leftarrow \epsilon\ell$ and apply updates as in Definition 2.2.

The SVAG algorithm describes a discrete trajectory that is a 1st-order approximation of the corresponding SDE (Definitions 4.1 and 4.3), thereby providing an efficient simulation of the SDEs.

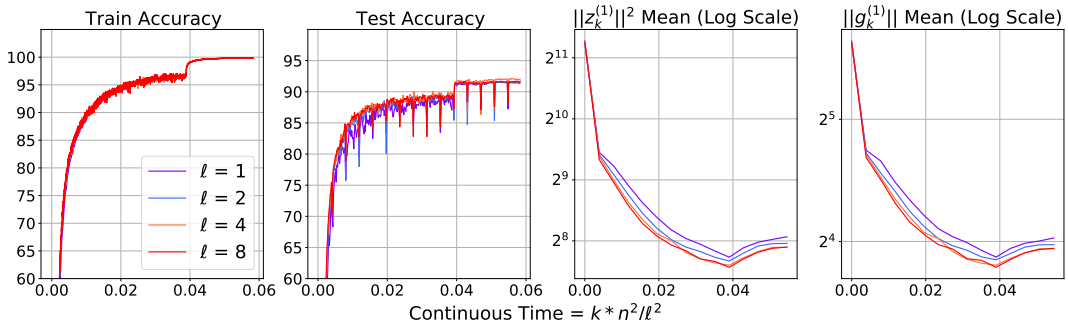


Figure 3: SVAG on the Adam trajectory when training ResNet-50 on CIFAR-10 matches the discrete trajectory ($\ell = 1$) on various test functions (see Appendix J for details). The closeness of the trajectories with respect to various test functions for different values of ℓ implies the SDE approximation (Definition 4.3) is a 1st-order weak approximation of Adam (Theorem 4.4).

Theorem 6.3 (SVAG algorithm approximates SDE)**.** *Assume the NGOS is well-behaved and satisfies the bounded moments condition (Definitions 2.5 and 2.7).*

1. *Let $\boldsymbol{X}_t$ be the state of the RMSprop SDE (Definition 4.1) with hyperparameters η, β, and ϵ. Let $\boldsymbol{x}_k$ be the state of the analogous discrete SVAG algorithm with hyperparameter ℓ.*
2. *Let $\boldsymbol{X}_t$ be the state of the Adam SDE (Definition 4.3) with hyperparameters η, β_1, β_2, and ϵ. Let $\boldsymbol{x}_k$ be the state of the analogous discrete SVAG algorithm with hyperparameter ℓ.*

In both 1 and 2, following holds for g and T as in Definition 2.4.

$$\max_{k=0,\ldots,\lfloor \ell^2 T/\eta^2 \rfloor} |\mathbb{E}g(\boldsymbol{x}_k) - \mathbb{E}g(\boldsymbol{X}_{k\eta^2/\ell^2})| \leq C\eta^2/\ell^2$$

Proof. The main idea of the proof is to show that the SVAG operator transforms the noise distribution of a well-behaved NGOS satisfying the bounded moments condition into one that is well-behaved and satisfies the bounded moments and the low skewness conditions (Lemma E.2). With these three conditions satisfied, we can directly apply Theorems 4.2 and 4.4 to complete the proof. □

Because the SDE scales time as $k = t/\eta^2$, we must run SVAG for ℓ^2 steps to match a single step of the discrete trajectories. Nevertheless, we note that in our setting and in [11], the approximation guarantee is strong enough for small ℓ, so this simulation is still more efficient than Euler-Maruyama.

Experiments. Figure 3 compares the Adam SVAG trajectories (Definition 6.2) up to $\ell = 8$ to the discrete one ($\ell = 1$) on CIFAR-10 [22] with ResNet-50 [29]. We use $\mathrm{Tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_k))$ and $\|\boldsymbol{g}_k\|$ as mathematically well-behaved test functions to test the approximation strength (see Definition 2.4). We also measure the train and test accuracies, even though they are not differentiable (and hence, not covered by the theory). The converged SVAG trajectories are close to the discrete ones under these test functions, suggesting the SDE approximations are applicable to realistic deep learning settings. Additional details and settings, including large language models, are in Appendix H.

7 Conclusion

We derive SDEs that are provable 1st-order approximations of the RMSprop and Adam trajectories, immediately yielding formal derivations of square root scaling rules: increase the learning rate by $\sqrt{\kappa}$ and adjust the adaptive hyperparameters when increasing batch size by κ. Experiments in the vision and language domains verify that applying these rules ensures that the values of several test functions, including test accuracy, are preserved. We furthermore design an efficient simulation for the SDEs, allowing us to directly validate the applicability of these SDEs to common vision and language settings. These SDEs can lead to a deeper understanding of how adaptivity and stochasticity impact optimization and generalization, and we hope to extend our results to formal identification of necessary and sufficient conditions for the approximation and its consequences to hold.

References

[1] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

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

[3] Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1731–1741. Curran Associates, Inc., 2017.

[4] Stanisław Jastrzębski, 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.

[5] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude., 2012.

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

[7] Chao Ma, Lei Wu, and Weinan E. A qualitative study of the dynamic behavior of adaptive gradient algorithms. *CoRR*, abs/2009.06125, 2020.

[8] Diego Granziol, Stefan Zohren, and Stephen Roberts. Learning rates as a function of batch size: A random matrix theory approach to neural network training, 2021.

[9] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2020.

[10] Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. *Journal of Machine Learning Research*, 20(40):1–47, 2019.

[11] Zhiyuan Li, Sadhika Malladi, and Sanjeev Arora. On the validity of modeling sgd with stochastic differential equations (sdes). *arXiv preprint arXiv:2102.12470*, 2021.

[12] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. *CoRR*, abs/1710.05941, 2017.

[13] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models?, 2020.

[14] Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5827–5837. PMLR, 09–15 Jun 2019.

[15] Pan Zhou, Jiashi Feng, Chao Ma, Caiming Xiong, Steven C. H. Hoi, and Weinan E. Towards theoretically understanding why SGD generalizes better than ADAM in deep learning. *CoRR*, abs/2010.05627, 2020.

[16] Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima, 2021.

[17] Zhiyuan Li, Kaifeng Lyu, and Sanjeev Arora. Reconciling modern deep learning with traditional optimization analyses: The intrinsic learning rate. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[18] Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel L. K. Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics, 2020.

[19] Zeke Xie, Xinrui Wang, Huishuai Zhang, Issei Sato, and Masashi Sugiyama. Adai: Separating the effects of adaptive learning rate and momentum inertia. *CoRR*, abs/2006.15815, 2020.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[22] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[24] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, 2015.

[25] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[26] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[27] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[28] Peter Izsak, Moshe Berchansky, and Omer Levy. How to train BERT with an academic budget. *CoRR*, abs/2104.07705, 2021.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[30] Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. Should you mask 15% in masked language modeling? *arXiv preprint arXiv:2202.08005*, 2022.

[31] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[32] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

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

[34] Jinghui Chen, Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao, and Quanquan Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021.

[35] Timothy Dozat. Incorporating nesterov momentum into adam. 2016.

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[37] Liangchen Luo, Yuanhao Xiong, and Yan Liu. Adaptive gradient methods with dynamic bound of learning rate. In *International Conference on Learning Representations*, 2019.

[38] Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in neural information processing systems*, 33:18795–18806, 2020.

[39] Jerry Ma and Denis Yarats. Quasi-hyperbolic momentum and adam for deep learning. In *International Conference on Learning Representations*, 2019.

[40] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[42] Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. *arXiv preprint arXiv:1910.07454*, 2019.

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] Conditions for our results are outlined in Definitions 2.5 to 2.7. In particular, we discuss the limited applicability of our work to cases in which the gradient noise is heavy-tailed, though our empirical success suggests that the gradient noise satisfies our assumptions in many realistic vision and language settings.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Our work is primarily theoretical in nature, but we discuss the broader impacts of our work in Appendix A.2.
   (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] Section 2 has a discussion of the assumptions on the gradient noise and the general requirements needed to be able to construct a continuous approximation of the discrete algorithms. Moreover, Section 4.1 motivates the assumption that the gradient noise dominates the gradient. We validate this assumption experimentally in Appendix G.1. Our assumptions do not stray from prior works on SDEs [10, 11].
   (b) Did you include complete proofs of all theoretical results? [Yes] Our proofs are written clearly in the appendix.
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] We include the code for the vision experiments in the supplementary material. For the NLP experiments, we use the code of [30].
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Appendix J contains the training details of all the experiments.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Due to computational constraints, we didn't include the error bars for all the experiments. However, we have made sure that the random seed for different runs in each experimental setting is the same.

(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] We ran our experiments on a cluster of 34 GPUs, where 24 are RTX 2080 GPUs and 10 are A5000 GPUs. Each experiment on CIFAR-10 required a single RTX 2080 GPU, each experiment on ImageNet required a single A5000 GPU, each pretraining experiment on GPT required a set of 4 RTX 2080 GPUs, each pretraining experiment on RoBERTa required a set of 8 RTX 2080 GPUs, and each finetuning experiment on RoBERTa required a single RTX 2080 GPU.

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? [Yes] We cite [30] for our language model experiments. We also cite the papers of all the deep learning architectures and datasets that we use for our experiments.

(b) Did you mention the license of the assets? [Yes] The github repository[5] of [30] has MIT license.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include our code for the vision experiments in the supplementary material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The data we used for our experiments are publicly available.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The data we used for our experiments are publicly available.

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]

[5] `https://github.com/princeton-nlp/DinkyTrain`