Stochastic Modified Equations and Adaptive Stochastic Gradient Algorithms

Qianxiao Li [1] **Cheng Tai** [2 3] **Weinan E** [2 3 4]

Abstract

We develop the method of stochastic modified equations (SME), in which stochastic gradient algorithms are approximated in the weak sense by continuous-time stochastic differential equations. We exploit the continuous formulation together with optimal control theory to derive novel adaptive hyper-parameter adjustment policies. Our algorithms have competitive performance with the added benefit of being robust to varying models and datasets. This provides a general methodology for the analysis and design of stochastic gradient algorithms.

1. Introduction

Stochastic gradient algorithms are often used to solve optimization problems of the form

$$\min_{x\in\mathbb{R}^d} \quad f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1}$$

where $f, f_i : \mathbb{R}^d \to \mathbb{R}$ for $i = 1, \dots, n$. In machine learning applications, f is typically the total loss function whereas each f_i represents the loss due to the i^{th} training sample. x is a vector of trainable parameters and n is the training sample size, which is typically very large.

Solving (1) using the standard gradient descent (GD) requires n gradient evaluations per step and is prohibitively expensive when $n \gg 1$. An alternative, the stochastic gradient descent (SGD), is to replace the full gradient ∇f by a sampled version, serving as an unbiased estimator. In its simplest form, the SGD iteration is written as

$$x_{k+1} = x_k - \eta \nabla f_{\gamma_k}(x_k), \tag{2}$$

where $k \geq 0$ and $\{\gamma_k\}$ are i.i.d uniform variates taking values in $\{1, 2, \cdots, n\}$. The step-size η is the learning rate. Unlike GD, SGD samples the full gradient and its computational complexity per iterate is independent of n. For this reason, stochastic gradient algorithms have become increasingly popular in large scale problems.

Many convergence results are available for SGD and its variants. However, most are upper-bound type results for (strongly) convex objectives, often lacking the precision and generality to characterize the behavior of algorithms in practical settings. This makes it harder to translate theoretical understanding into algorithm analysis and design.

In this work, we address this by pursuing a different analytical direction. We derive continuous-time stochastic differential equations (SDE) that can be understood as weak approximations (i.e. approximations in distribution) of stochastic gradient algorithms. These SDEs contain higher order terms that vanish as $\eta \to 0$, but at finite and small η they offer much needed insight of the algorithms under consideration. In this sense, our framework can be viewed as a stochastic parallel of the method of *modified equations* in the analysis of classical finite difference methods (Noh & Protter, 1960; Daly, 1963; Hirt, 1968; Warming & Hyett, 1974). For this reason, we refer to these SDEs as *stochastic modified equations* (SME). Using the SMEs, we can quantify, in a precise and general way, the leading-order dynamics of the SGD and its variants. Moreover, the continuous-time treatment allows the application of optimal control theory to study the problems of adaptive hyper-parameter adjustments. This gives rise to novel adaptive algorithms and perhaps more importantly, a general methodology for understanding and improving stochastic gradient algorithms.

Notation. We distinguish sequential and dimensional indices by writing a bracket around the latter, e.g. $x_{k,(i)}$ is the i^{th} coordinate of the vector x_k, the k^{th} SGD iterate.

2. Stochastic Modified Equations

We now introduce the SME approximation. Background materials on SDEs are found in Supplementary Materials

[1]Institute of High Performance Computing, Singapore [2]Peking University, Beijing, China [3]Beijing Institute of Big Data Research, Beijing, China [4]Princeton University, Princeton, NJ, USA. Correspondence to: Qianxiao Li <liqix@ihpc.a-star.edu.sg>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

(SM) B and references therein. First, rewrite the SGD iteration rule (2) as

$$x_{k+1} - x_k = -\eta \nabla f(x_k) + \sqrt{\eta} V_k, \tag{3}$$

where $V_k = \sqrt{\eta}(\nabla f(x_k) - \nabla f_{\gamma_k}(x_k))$ is a d-dimensional random vector. Conditioned on x_k, V_k has mean 0 and covariance matrix $\eta\Sigma(x_k)$ with

$$\Sigma(x) = \frac{1}{n}\sum_{i=1}^{n}(\nabla f(x) - \nabla f_i(x))(\nabla f(x) - \nabla f_i(x))^T. \tag{4}$$

Now, consider the Stochastic differential equation

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t, \quad X_0 = x_0, \tag{5}$$

whose Euler discretization $X_{k+1} = X_k + \Delta t b(X_k) + \sqrt{\Delta t}\sigma(X_k)Z_k$, $Z_k \sim \mathcal{N}(0, I)$ resembles (3) if we set $\Delta t = \eta$, $b \sim -\nabla f$ and $\sigma \sim (\eta\Sigma)^{1/2}$. Then, we would expect (5) to be an approximation of (2) with the identification $t = k\eta$. It is now important to discuss the precise meaning of "an approximation". The noises that drive the paths of SGD and SDE are independent processes, hence we must understand approximations in the *weak sense*.

Definition 1. *Let $0 < \eta < 1$, $T > 0$ and set $N = \lfloor T/\eta \rfloor$. Let G denote the set of functions of polynomial growth, i.e. $g \in G$ if there exists constants $K, \kappa > 0$ such that $|g(x)| < K(1 + |x|^\kappa)$. We say that the SDE (5) is an order α weak approximation to the SGD (2) if for every $g \in G$, there exists $C > 0$, independent of η, such that for all $k = 0, 1, \ldots, N$,*

$$|\mathbb{E}g(X_{k\eta}) - \mathbb{E}g(x_k)| < C\eta^\alpha.$$

The definition above is standard in numerical analysis of SDEs (Milstein, 1995; Kloeden & Platen, 2011). Intuitively, weak approximations are close to the original process not in terms of individual sample paths, but their distributions. We now state informally the approximation theorem.

Informal Statement of Theorem 1. *Let $T > 0$ and define $\Sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ by (4). Assume f, f_i are Lipschitz continuous, have at most linear asymptotic growth and have sufficiently high derivatives belonging to G. Then,*

(i) The stochastic process X_t, $t \in [0, T]$ satisfying

$$dX_t = -\nabla f(X_t)dt + (\eta\Sigma(X_t))^{\frac{1}{2}}dW_t, \tag{6}$$

is an order 1 weak approximation of the SGD.

(ii) The stochastic process X_t, $t \in [0, T]$ satisfying

$$dX_t = -\nabla(f(X_t) + \tfrac{\eta}{4}|\nabla f(X_t)|^2)dt + (\eta\Sigma(X_t))^{\frac{1}{2}}dW_t \tag{7}$$

is an order 2 weak approximation of the SGD.

The full statement, proof and numerical verification of Thm. 1 is given in SM. C. We hereafter call equations (6) and (7) *stochastic modified equations* (SME) for the SGD iterations (2). We refer to the second order approximation (7) for exact calculations in Sec. 3 whereas for simplicity, we use the first order approximation (6) when discussing acceleration schemes in Sec. 4, where the order of accuracy is less important.

Thm. 1 allows us to use the SME to deduce distributional properties of the SGD. This result differs from usual convergence studies in that it describes dynamical behavior and is derived without convexity assumptions on f or f_i. In the next section, we use the SME to deduce some dynamical properties of the SGD.

3. The Dynamics of SGD

3.1. A Solvable SME

We start with a case where the SME is exactly solvable. Let $n = 2$, $d = 1$ and set $f(x) = x^2$ with $f_1(x) = (x-1)^2 - 1$ and $f_2(x) = (x+1)^2 - 1$. Then, the SME (7) for the SGD iterations on this objective is (see SM. D.1)

$$dX_t = -2(1+\eta)X_t dt + 2\sqrt{\eta}dW_t,$$

with $X_0 = x_0$. This is the well-known Ornstein-Uhlenbeck process (Uhlenbeck & Ornstein, 1930), which is exactly solvable (see SM. B.3), yielding the Gaussian distribution

$$X_t \sim \mathcal{N}(x_0 e^{-2(1+\eta)t}, \tfrac{\eta}{1+\eta}(1 - e^{-4(1+\eta)t})).$$

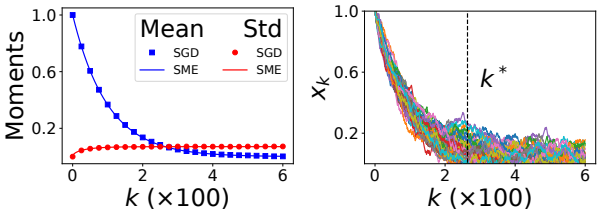
We observe that $\mathbb{E}X_t = x_0 e^{-2(1+\eta)t}$ converges exponentially to the optimum $x = 0$ with rate $-2(1+\eta)$ but $\mathrm{Var}X_t = \eta\left(1 - e^{-4(1+\eta)t}\right)/(1+\eta)$ increases from 0 to an asymptotic value of $\eta/(1+\eta)$. The separation t^* between the descent phase and the fluctuations phase is given by $\mathbb{E}X_{t^*} = \sqrt{\mathrm{Var}X_{t^*}}$, whose solution is

$$t^* = \tfrac{1}{4(1+\eta)}\log(1 + \tfrac{\eta+1}{\eta}x_0^2)$$

For $t < t^*$, descent dominates and when $t > t^*$, fluctuation dominates. This two-phase behavior is known for convex cases via error bounds (Moulines, 2011; Needell et al., 2014). Using the SME, we obtained a precise characterization of this behavior, including an exact expression for t^*. In Fig. 1, we verify the SME predictions regarding the mean, variance and the two-phase behavior.

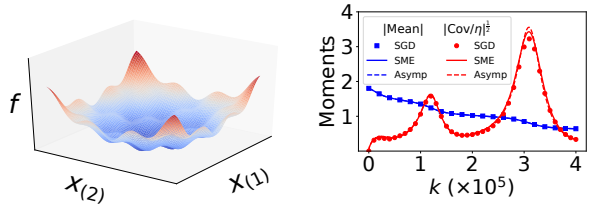
3.2. Stochastic Asymptotic Expansion

In general, we cannot expect to solve the SME exactly, especially for $d > 1$. However, observe that the noise terms in the SMEs (6) and (7) are $\mathcal{O}(\eta^{1/2})$. Hence, we can write X_t as an asymptotic series $X_t = X_{0,t} + \sqrt{\eta}X_{1,t} + \ldots$



(a) (b)

Figure 1. Comparison of the SME predictions vs SGD for the simple quadratic objective. We set $x_0 = 1$, η =5e-3. (a) The predicted mean and standard deviations agree well with the empirical moments of the SGD, obtained by averaging 5e3 runs. (b) 50 sample SGD paths the predicted transition time $k^* = t^*/\eta$. We observe that k^* corresponds to the separation of descent and fluctuating regimes for typical sample paths.



(a) (b)

Figure 2. Comparison of the moments of SGD iterates with the SME and its asymptotic approximation (Asymp, Eq. 8) for the non-convex objective with $\delta = 0.2$ and $\epsilon = 0.1$. The landscape is shown in (a). In (b), we plot the magnitude of the mean and the covariance matrix for the SGD, SME and Asymp. We take η =1e-4 and $x_0 = (1, 1.5)$. All moments are obtained by sampling over 1e3 runs (the SME and Asymp are integrated numerically). We observe a good agreement.

where each $X_{j,t}$ is a stochastic process with initial condition $X_{0,0} = x_0$ and $X_{j,0} = 0$ for $j \geq 1$. We substitute this into the SME and expand in orders of $\eta^{1/2}$ and equate the terms of the same order to get equations for $X_{j,t}$ for $j \geq 0$. This procedure is justified rigorously in Freidlin et al. (2012). We obtain to leading order (see SM. B.5),

$$X_t \sim \mathcal{N}(X_{0,t}, \eta S_t), \tag{8}$$

where $X_{0,t}$ solves $\dot{X}_{0,t} = -\nabla f(X_{0,t}), X_{0,0} = x_0$ and $\dot{S}_t = -S_t H_t - H_t S_t + \Sigma_t$, where $H_t = Hf(X_{0,t})$, with Hf denoting the Hessian of f, and $\Sigma_t = \Sigma(X_{0,t})$, $S_0 = 0$. It is then possible to deduce the dynamics of the SGD. For example, there is generally a transition between descent and fluctuating regimes. S_t has a steady state (assuming it is asymptotically stable) with $|S_\infty| \sim |\Sigma_\infty|/|H_\infty|$. This means that one should expect a fluctuating regime where the covariance of the SGD is of order $\mathcal{O}(\eta|\Sigma_\infty|/|H_\infty|)$. Preceding this fluctuating regime is a descent regime governed by the gradient flow.

We validate our approximations on a non-convex objective. Set $d = 2$, $n = 3$ with the sample objectives $f_1(x) = x_{(1)}^2$, $f_2(x) = x_{(2)}^2$ and $f_3(x) = \delta \cos(x_{(1)}/\epsilon) \cos(x_{(2)}/\epsilon)$. In Fig. 2(a), we plot f for $\epsilon = 0.1, \delta = 0.2$, showing the complex landscape. In Fig. 2(b), we compare the SGD moments $|\mathbb{E}(x_k)|$ and $|\mathrm{Cov}(x_k)|$ with predictions of the SME and its asymptotic approximation (8). We observe that our approximations indeed hold for this objective.

4. Adaptive Hyper-parameter Adjustment

We showed in the previous section that the SME formulation help us better understand the precise dynamics of the SGD. The natural question is how this can translate to designing practical algorithms. In this section, we exploit the continuous-time nature of our framework to derive adaptive learning rate and momentum parameter adjustment policies. These are particular illustrations of a general methodology to analyze and improve upon SGD variants. We will focus on the one dimensional case $d = 1$, and subsequently apply the results to high dimensional problems by local diagonal approximations.

4.1. Learning Rate

4.1.1. Optimal Control Formulation

1D SGD iterations with learning rate adjustment can be written as

$$x_{k+1} = x_k - \eta u_k f'(x_k), \tag{9}$$

where $u_k \in [0, 1]$ is the adjustment factor and η is the maximum allowed learning rate. The corresponding SME for (9) is given by (SM. D.1)

$$dX_t = -u_t f'(X_t)dt + u_t \sqrt{\eta \Sigma(X_t)} dW_t, \tag{10}$$

where $u_t \in [0, 1]$ is now the continuous time analogue of the adjustment factor u_k with the usual identification $t = k\eta$. The effect of learning rate adjustment on the dynamics of SGD is clear. Larger u_k results in a larger drift term in the SME and hence faster initial descent. However, the same factor is also multiplied to the noise term, causing greater asymptotic fluctuations. The optimal learning rate schedule must balance of these two effects. The problem can therefore be posed as follows: given f, f_i, how can we best choose a schedule or policy for adjusting the learning rate in order to minimize $\mathbb{E}f$ at the end of the run? More precisely, this can be cast as an optimal control problem[1]

$$\min_u \mathbb{E} f(X_T) \text{ subject to (10)},$$

[1] See SM. E for a brief overview of optimal control theory.

where the time-dependent function u is minimized over an admissible control set to be specified. To make headway analytically, we now turn to a simple quadratic objective.

4.1.2. Optimal Control of the Learning Rate

Consider the objective $f(x) = \frac{1}{2}a(x-b)^2$ with $a, b \in \mathbb{R}$. Moreover, we assume the f_i's are such that $\Sigma(x) = \Sigma > 0$ is a positive constant. The SME is then

$$dX_t = -au_t(X_t - b)dt + u_t\sqrt{\eta\Sigma}dW_t. \tag{11}$$

Now, assume u take values in the non-random control set containing all Borel-measurable functions from $[0, T]$ to $[0, 1]$. Defining $m_t = \mathbb{E}f(X_t)$, and applying Itô formula to (11), we have

$$\dot{m}_t = -2au_tm_t + \tfrac{1}{2}a\eta\Sigma u_t^2. \tag{12}$$

Hence, we may now recast the control problem as

$$\min_{u:[0,T]\to[0,1]} m_T \text{ subject to (12)}.$$

This problem can solved by dynamic programming, using the Hamilton-Jacobi-Bellman equation (Bellman, 1956). We obtain the optimal control policy (SM. E.3)

$$u_t^* = \begin{cases} 1 & a \le 0, \\ \min(1, \frac{2m_t}{\eta\Sigma}) & a > 0. \end{cases} \tag{13}$$

This policy is of *feed-back* form since it depends on the current value of the controlled variable m_t. Let us interpret the solution. First, if $a < 0$ we always set the maximum learning rate $u_t = 1$. This makes sense because we have a concave objective where symmetrical fluctuations about any point x results in a lower average value of $f(x)$. Hence, not only do high learning rates improve descent, the high fluctuations that accompany it also lowers $\mathbb{E}f$. Next, For the convex case $a > 0$, the solution tells us that when the objective value is large compared to variations in the gradient, we should use the maximum learning rate. When the objective decreases sufficiently, fluctuations will dominate and hence we should lower the learning rate according to the feed-back policy $u_t = 2m_t/\eta\Sigma$.

With the policy (13), we can solve (12) and plug the solution for m_t back into (13) to obtain the annealing schedule

$$u_t^* = \begin{cases} 1 & a \le 0 \text{ or } t \le t^*, \\ \frac{1}{1+a(t-t^*)} & a > 0 \text{ and } t > t^*, \end{cases}$$

where $t^* = (1/2a)\log(4m_0/\eta\Sigma - 1)$. Note that by putting $a = 2, b = 0, \Sigma = 4$, for small η, this expression agrees with the transition time (8) between descent and fluctuating phases for the SGD dynamics considered in Sec. 3.1. Thus, this annealing schedule says that maximum learning rate should be used for descent phases, whereas $\sim 1/t$ decay on learning rate should be applied after onset of fluctuations. Our annealing result agree asymptotically with the commonly studied annealing schedules (Moulines, 2011; Shamir & Zhang, 2013), but the difference is that we suggest maximum learning rate before the onset of fluctuations. Of course, the key limitation is that our result is only valid for this particular objective. This naturally brings us to the next question: how does one apply the optimal control results to general objectives?

4.1.3. Application to General Objectives

Now, we turn to the setting where $d > 1$ and f, f_i are not necessarily quadratic. The most important result in Sec. 4.1.2 is the feed-back control law (13). To apply it, we make a *local diagonal-quadratic assumption*: we assume that for each $x \in \mathbb{R}^d$, there exists $a_{(i)}, b_{(i)} \in \mathbb{R}$ so that $f(x) \approx \frac{1}{2}\sum_{i=1}^d a_{(i)}(x_{(i)} - b_{(i)})^2$ holds locally in x. We also assume $\Sigma(x) \approx \text{diag}\{\Sigma_{(1)}, \ldots, \Sigma_{(d)}\}$ where each $\Sigma_{(i)}$ is locally constant. By considering a separate learning rate scale $u_{(i)}$ for each trainable dimension, the control problem decouples to d separate problems of the form considered in Sec. 4.1.2. And hence, we may set $u_{(i)}^*$ element-wise according to the policy (13).

Since we only assume that the diagonal-quadratic assumption holds locally, the terms $a_{(i)}$, $b_{(i)}$, $\Sigma_{(i)}$ and $m_{(i)} \approx \frac{1}{2}a_{(i)}(x_{(i)} - b_{(i)})^2$ must be updated on the fly. There are potentially many methods for doing so. The approach we take exploits the linear relationship $\nabla f_{(i)} \approx a_{(i)}(x_{(i)} - b_{(i)})$. Consequently, we may estimate $a_{(i)}, b_{(i)}$ via linear regression on the fly: for each dimension, we maintain exponential moving averages (EMA) $\{\overline{g}_{k,(i)}, \overline{g^2}_{k,(i)}, \overline{x}_{k,(i)}, \overline{x^2}_{k,(i)}, \overline{xg}_{k,(i)}\}$ where $g_{k,(i)} = \nabla f_{\gamma_k}(x_k)_{(i)}$. For example, $\overline{g}_{k+1,(i)} = \beta_{k,(i)}\overline{g}_{k,(i)} + (1 - \beta_{k,(i)})g_{k,(i)}$. The EMA decay parameter $\beta_{k,(i)}$ controls the effective averaging window size. We adaptively adjust it so that it is small when gradient variations are large, and vice versa. We employ the heuristic $\beta_{k+1,(i)} = (\overline{g^2}_{k,(i)} - \overline{g}^2_{k,(i)})/\overline{g^2}_{k,(i)}$. This is similar to the approach in Schaul et al. (2013). We also clip each $\beta_{k+1,(i)}$ to $[\beta_{\text{min}}, \beta_{\text{max}}]$ to improve stability. Here, we use $[0.9, 0.999]$ for all experiments, but we checked that performance is insensitive to these values. We can now compute $a_{k,(i)}, b_{k,(i)}$ by the ordinary-least-squares formula and $\Sigma_{k,(i)}$ as the variance of the gradients:

$$\begin{aligned} a_{k,(i)} &= \frac{\overline{gx}_{k,(i)} - \overline{g}_{k,(i)}\overline{x}_{k,(i)}}{\overline{x^2}_{k,(i)} - \overline{x}^2_{k,(i)}}, \\ b_{k,(i)} &= \overline{x}_{k,(i)} - \frac{\overline{g}_{k,(i)}}{a_{k,(i)}}, \\ \Sigma_{k,(i)} &= \overline{g^2}_{k,(i)} - \overline{g}^2_{k,(i)}. \end{aligned} \tag{14}$$

Algorithm 1 controlled SGD (cSGD)

Hyper-parameters: η, u_0
Initialize x_0; $\beta_{0,(i)} = 0.9\ \forall i$
for $k = 0$ **to** (#iterations $- 1$) **do**
  Compute sample gradient $\nabla f_{\gamma_k}(x_k)$
  for $i = 1$ **to** d **do**
    Update EMA $\{\overline{g}_{k,(i)}, \overline{g^2}_{k,(i)}, \overline{x}_{k,(i)}, \overline{x^2}_{k,(i)}, \overline{xg}_{k,(i)}\}$ with decay parameter $\beta_{k,(i)}$
    Compute $a_{k,(i)}$, $b_{k,(i)}$, $\Sigma_{k,(i)}$ using (14)
    Compute $u^*_{k,(i)}$ using (15)
    $\beta_{k+1,(i)} = (\overline{g^2}_{k,(i)} - \overline{g}^2_{k,(i)})/\overline{g^2}_{k,(i)}$ and clip
    $u_{k+1,(i)} = \beta_{k,(i)} u_{k,(i)} + (1 - \beta_{k,(i)}) u^*_{k,(i)}$
    $x_{k+1,(i)} = x_{k,(i)} - \eta u_{k,(i)} \nabla f_{\gamma_k}(x_k)_{(i)}$
  end for
end for

This allows us to estimate the policy (13) as

$$u^*_{k,(i)} = \begin{cases} 1 & a_{k,(i)} \leq 0, \\ \min(1, \frac{a_{k,(i)}(\overline{x}_{k,(i)} - b_{k,(i)})^2}{\eta \Sigma_{k,(i)}}) & a_{k,(i)} > 0. \end{cases} \tag{15}$$

for $i = 1, 2, \ldots, d$. Since quantities are computed from exponentially averaged sources, we should also update our learning rate policy in the same way. The algorithm is summarized in Alg. 1. Due to its optimal control origin, we hereafter call this algorithm the *controlled SGD* (cSGD)

Remark 1. *Alg. 1 can similarly be applied to mini-batch SGD. Let the batch-size be M, which reduces the covariance by M times and so η in the SME is replaced by η/M. However, at the same time estimating Σ_k from mini-batch gradient sample variances will underestimate $\Sigma(x_k)$ by a factor of M. Thus the product $\eta\Sigma_k$ remains unchanged and Alg. 1 can be applied with no changes.*

Remark 2. *The additional overheads in cSGD are from maintaining exponential averages and estimating a_k, b_k, Σ_k on the fly with the relevant formulas. These are $\mathcal{O}(d)$ operations and hence scalable. Our current rough implementation runs $\sim 40 - 60\%$ slower per epoch than the plain SGD. This is expected to be improved by optimization, parallelization or updating quantities less frequently.*

4.1.4. Performance on Benchmarks

Let us test cSGD on common deep learning benchmarks. We consider three different models. M0: a fully connected neural network with one hidden layer and ReLU activations, trained on the MNIST dataset (LeCun et al., 1998); C0: a fully connected neural network with two hidden layers and Tanh activations, trained on the CIFAR-10 dataset (Krizhevsky & Hinton, 2009); C1: a convolution network with four convolution layers and two fully connected layers also trained on CIFAR-10. Model details are found in SM. F.1. In Fig. 3, we compare the performance of cSGD with Adagrad (Duchi et al., 2011) and Adam (Kingma & Ba, 2015) optimizers. We illustrate in particular their sensitivity to different learning rate choices by performing a log-uniform random search over three orders of magnitude. We observe that cSGD is robust to different initial and maximum learning rates (provided the latter is big enough, e.g. we can take $\eta = 1$ for all experiments) and changing network structures, while obtaining similar performance to well-tuned versions of the other methods (see also Tab. 1). In particular, notice that the best learning rates found for Adagrad and Adam generally differ for different neural networks. On the other hand, many values can be used for cSGD with little performance loss. For brevity we only show the test accuracies, but the training accuracies have similar behavior (see SM. F.5).

4.2. Momentum Parameter

Another practical way of speeding up the plain SGD is to employ momentum updates - an idea dating back to deterministic optimization (Polyak, 1964; Nesterov, 1983; Qian, 1999). However, the stochastic version has important differences, especially in regimes where sampling noise dominates. Nevertheless, provided that the momentum parameter is well-tuned, the momentum SGD (MSGD) is very effective in speeding up convergence, particularly in early stages of training (Sutskever et al., 2013).

Selecting an appropriate momentum parameter is important in practice. Typically, generic values (e.g. 0.9, 0.99) are suggested without fully elucidating their effect on the SGD dynamics. In this section, we use the SME framework to analyze the precise dynamics of MSGD and derive effective adaptive momentum parameter adjustment policies.

4.2.1. SME for MSGD

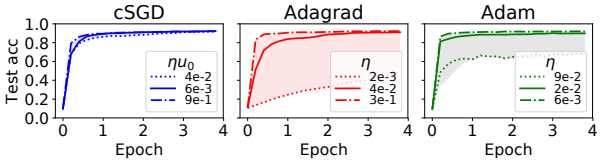
The SGD with momentum can be written as the following coupled updates

$$\begin{aligned} v_{k+1} &= \mu v_k - \eta f'_{\gamma_k}(x_k), \\ x_{k+1} &= x_k + v_{k+1}. \end{aligned} \tag{16}$$

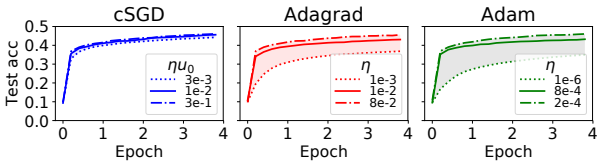
The parameter μ is the momentum parameter taking values in the range $0 \leq \mu \leq 1$. Intuitively, the momentum term v_k remembers past update directions and pushes along x_k, which may otherwise slow down at e.g. narrow parts of the landscape. The corresponding SME is now a coupled SDE

$$\begin{aligned} dV_t &= (-\eta^{-1}(1-\mu)V_t - f'(X_t))dt + (\eta\Sigma(X_t))^{\frac{1}{2}} dW_t, \\ dX_t &= \eta^{-1} V_t dt. \end{aligned} \tag{17}$$

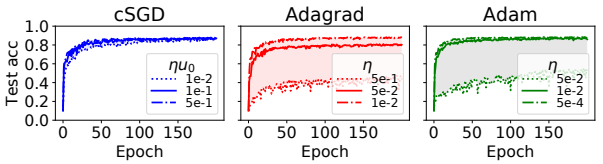
This can be derived by comparing (16) with the Euler discretization scheme of (17) and matching moments. Details can be found in SM. D.3.



(a) M0 (fully connected NN, MNIST)



(b) C0 (fully connected NN, CIFAR-10)



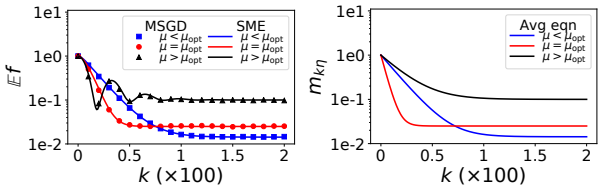
(c) C1 (CNN, CIFAR-10)

Figure 3. cSGD vs Adagrad and Adam for different models and datasets, with different hyper-parameters. For M0, we perform log-uniform random search with 50 samples over intervals: cSGD: u_0 ∈[1e-2,1], η ∈[1e-1,1]; Adagrad: η ∈[1e-3,1]; Adam: η ∈[1e-4,1e-1]. For C0, we perform same search over intervals: cSGD: u_0 ∈[1e-2,1], η ∈[1e-1,1]; Adagrad: η ∈[1e-3,1]; Adam: η ∈[1e-6,1e-3]. We average the resulting learning curves for each choice over 10 runs. For C1, due to long training times we choose 5 representative learning rates for each method. cSGD: η ∈{1e-2,5e-2,1e-1,5e-1,1}, $u_0 = 1$; Adagrad: η ∈{1e-3,5e-3,1e-2,5e-2,1e-1}; Adam: η ∈{5e-4,1e-3,1e-2,2e-2,5e-2}. One sample learning curve is generated for each choice. In all cases, we use mini-batches of size 128. We evaluate the resulting learning curves by the area-under-curve. The worst, median and best learning curves are shown as dotted, solid, and dot-dashed lines respectively. The shaded areas represent the distribution of learning curves for all searched values. We observe that cSGD is relatively robust with respect to initial/maximum learning rates and the network structures, and requires little tuning while having comparable performance to well-tuned versions of the other methods (see Tab. 1). This holds across different models and datasets.

4.2.2. The Effect of Momentum

As in Sec. 4.1, we take the prototypical example $f(x) = \frac{1}{2}a(x-b)^2$ with Σ constant and study the effect of incorporating momentum updates. Define $M_t = (\mathbb{E}f(X_t), \mathbb{E}V_t^2, \mathbb{E}V_t f'(X_t)) \in \mathbb{R}^3$. By applying Itô formula to (17), we obtain the ODE system

$$\dot{M}_t = A(\mu)M_t + B,$$
$$A(\mu) = \begin{pmatrix} 0 & 0 & a/\eta \\ 0 & -2(1-\mu)/\eta & -2 \\ -2 & 1/\eta & -(1-\mu)/\eta \end{pmatrix}, \; B = \begin{pmatrix} 0 \\ \eta\Sigma \\ 0 \end{pmatrix}. \quad (18)$$



(a) (b)

Figure 4. (a) Comparison of the SME prediction (18) with SGD for the same quadratic example in Sec. 3.1, which has $a = 2$, $b = 0$ and $\Sigma = 4$. We set η =5e-3 so that $\mu_{\text{opt}} = 0.8$. We plot the mean of f averaged over 1e5 SGD runs against the SME predictions for $\mu = 0.65, 0.8, 0.95$. We observe that in all cases the approximation is accurate. In particular, the SME correctly predicts the effect of momentum: $\mu = \mu_{\text{opt}}$ gives the best average initial descent rate, $\mu > \mu_{\text{opt}}$ causes oscillatory behavior, and increasing μ generally increases asymptotic fluctuations. (b) The dynamics of averaged equation (20), which serves as an approximation of the solution of the full SME moment equation (18).

If $a < 0$, $A(\mu)$ has a positive eigenvalue and hence M_t diverges exponentially. Since f is negative, its value must then decrease exponentially for all μ, and the descent rate is maximized at $\mu = 1$. The more interesting case is when $a > 0$. Instead of solving (18), we observe that all eigenvalues of $A(\mu)$ have negative real parts as long as $\mu < 1$. Therefore, M_t has an exponential decay dominated by $|\mathcal{R}\lambda(\mu)|$, where $\mathcal{R}$ denotes real part and $\lambda(\mu) = -\frac{1}{\eta}[(1-\mu) - \sqrt{(1-\mu)^2 - 4a\eta}]$ is the eigenvalue with the least negative real part. Observe that the descent rate $|\mathcal{R}\lambda(\mu)|$ is maximized at

$$\mu_{\text{opt}} = \max(1 - 2\sqrt{a\eta}, 0) \quad (19)$$

and when $\mu > \mu_{\text{opt}}$, λ becomes complex. Also, from (18) we have $M_t \to M_\infty = -A(\mu)^{-1}B = \left(\frac{\eta\Sigma}{4(1-\mu)} \; \frac{\eta^2\Sigma}{2(1-\mu)} \; 0\right)$, provided the steady state is stable. The role of momentum in this problem is now clear. To leading order in η we have $\lambda(\mu) \sim -2a/(1-\mu)$ for $\mu \le \mu_{\text{opt}}$. Hence, any non-zero momentum will improve the initial convergence rate. In fact, the choice μ_{opt} is optimal and above it, oscillations set in because of a complex λ. At the same time, increasing momentum also causes increment in eventual fluctuations, since $|M_\infty| = \mathcal{O}((1-\mu)^{-1})$. In Fig. 4(a), we demonstrate the accuracy of the SME prediction (18) by comparing MSGD iterations. Armed with an intuitive understanding of the effect of momentum, we can now use optimal control to design policies to adapt the momentum parameter.

4.2.3. Optimal Control of the Momentum Parameter

For $a < 0$, we have discussed previously that $\mu = 1$ maximizes the descent rate and fluctuations generally help de-

crease concave functions. Thus, the optimal control is always $\mu = 1$. The non-trivial case is when $a > 0$. Due to its bi-linearity, directly controlling (18) leads to bang-bang type solutions[2] that are rarely feed-back laws (Pardalos & Yatsenko, 2010) and thus difficult to apply in practice. Instead, we notice that the descent rate is dominated by $\mathcal{R}\lambda(\mu)$, and the leading order asymptotic fluctuations is $\eta\Sigma/(4(1-\mu))$, hence we may consider

$$\dot{m}_t = \mathcal{R}\lambda(\mu)(m_t - m_\infty(\mu)) \tag{20}$$

where $m_t \in \mathbb{R}$ and $m_\infty(\mu) = \eta\Sigma/(4(1-\mu))$ is the leading order estimate of $|M_\infty|$. Equation (20) can be understood as the approximate evolution, in an averaged sense, of the magnitude of M_t. Fig. 4(b) shows that (20) is a reasonable approximation of the dynamics of MSGD. This allows us to pose the optimal control problem on the momentum parameter as

$$\min_{\mu:[0,T]\to[0,1]} m_T \text{ subject to (20)},$$

with $\mu = \mu_t$. Solving this control problem yields the (approximate) feed-back policy (SM. E.4)

$$\mu_t^* = \begin{cases} 1 & a \le 0, \\ \min(\mu_{\text{opt}}, \max(0, 1 - \frac{\eta\Sigma}{4m_t})) & a > 0, \end{cases} \tag{21}$$

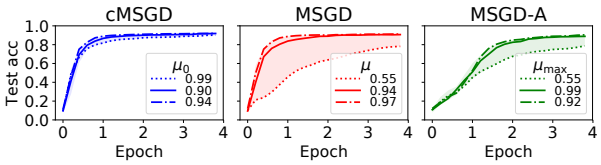
with μ_{opt} given in (19). This says that when far from optimum (m_t large), we set $\mu = \mu_{\text{opt}}$ which maximizes average descent rate. When $m_t/\eta\Sigma \sim \sqrt{a\eta}$, fluctuations set in and we lower μ.

As in Sec. 4.1.3, we turn the control policy above into a generally applicable algorithm by performing local diagonal-quadratic approximations and estimating the relevant quantities on the fly. The resulting algorithm is mostly identical to Alg. 1 except we now use (21) to update $\mu_{k,(i)}$ and SGD updates are replaced with MSGD updates (see S.M. F.4 for the full algorithm). We refer to this algorithm as the *controlled momentum SGD* (cMSGD).

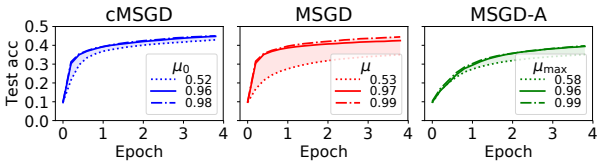
4.2.4. Performance on Benchmarks

We apply cMSGD to the same three set-ups in Sec. 4.1.4, and compare its performance to the plain Momentum SGD with fixed momentum parameters (MSGD) and the annealing schedule suggested in (Sutskever et al., 2013), with $\mu_k = \min(1 - 2^{-1-\log_2(\lfloor k/250 \rfloor + 1)}, \mu_{\max})$ (MSGD-A). In Fig. 5, we perform a log-uniform search over the hyper-parameters μ_0, μ and $\mu_{\max}$. We see that cMSGD achieves superior performance to MSGD and MSGD-A (see Tab. 1),

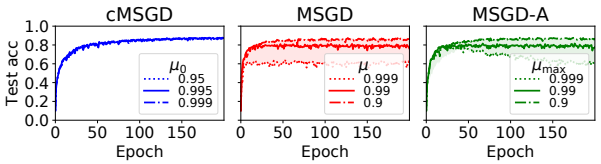
[2]Bang-bang solutions are control solutions lying on the boundary of the control set and abruptly jumps among the boundary values. For example, in this case it jumps between $\mu = 0$ and $\mu = 1$ repeatedly.



(a) M0 (fully connected NN, MNIST)



(b) C0 (fully connected NN, CIFAR-10)



(c) C1 (CNN, CIFAR-10)

Figure 5. cMSGD vs MSGD and MSGD-A on the same three models. We set η =1e-2 for M0 and η =1e-3 for C0, C1. For M0 and C0, we perform a log-uniform random search for $1-\mu_0$ and $1-\mu$ in [5e-3,5e-1]. For C1, we sample $\mu_0, \mu, \mu_{\max} \in \{0.9, 0.95, 0.99, 0.995, 0.999\}$. The remaining set-up is identical to that in Fig. 3. Again, we observe that cMSGD is an adaptive scheme that is robust to varying hyper-parameters and network structures, and out-performs MSGD and MSGD-A.

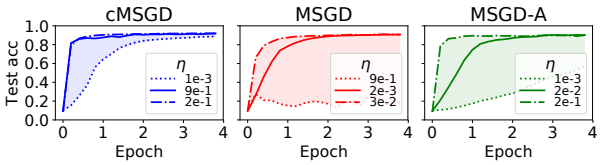
Table 1. Best average test accuracy found by random/grid search.

	cSGD	Adagrad	Adam	cMSGD	MSGD	MSGD-A
M0	**0.925**	0.923	0.924	**0.924**	0.914	0.908
C0	**0.461**	0.457	0.460	**0.461**	0.453	0.446
C1	0.875	0.878	**0.881**	**0.876**	0.868	0.869

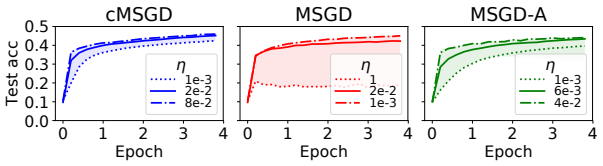
especially when the latter has badly tuned $\mu, \mu_{\max}$. Moreover, it is insensitive to the choice of initial μ_0. Just like cSGD, this holds across changing network structures. Further, cMSGD also adapts to other hyper-parameter variations. In Fig. 6, we take tuned $\mu, \mu_{\max}$ (and any μ_0) and vary the learning rate η. We observe that cMSGD adapts to the new learning rates whereas the performance of MSGD and MSGD-A deteriorates and $\mu, \mu_{\max}$ must be re-tuned to obtain reasonable accuracy. In fact, it is often the case that MSGD and MSGD-A diverge when η is large, whereas cMSGD remains stable.

5. Related Work

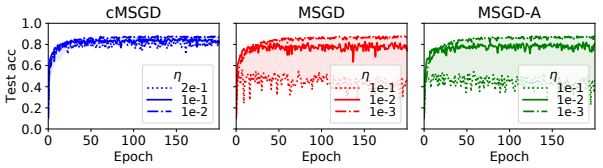
Classical bound-type convergence results for SGD and variants include Moulines (2011); Shamir & Zhang (2013);



(a) M0 (fully connected NN, MNIST)



(b) C0 (fully connected NN, CIFAR-10)



(c) C1 (CNN, CIFAR-10)

Figure 6. Comparing the sensitivity of cMSGD, MSGD and MSGD-A to different learning rates. The set-up is same as that in Fig. 5 except that for MSGD and MSGD-A, we now fix $\mu, \mu_{\max}$ to be the best values found in Fig. 5 for each experiment, but we vary the learning rate in the ranges: M0 and C0: $\eta \in$[1e-3,1], C1: $\eta \in$\{1e-3,5e-3,1e-2,2e-2,1e-1\}. For cMSGD, we saw from Fig. 5 that the value of μ_0 is mostly inconsequential, so we simply set $\mu_0 = 0$ and vary η in the same ranges. We observe that the unlike MSGD and MSGD-A, cMSGD is generally robust to changing learning rates and this further confirms its adaptive properties.

Bach & Moulines (2013); Needell et al. (2014); Xiao & Zhang (2014); Shalev-Shwartz & Zhang (2014). Our approach differs in that we obtain precise, albeit only distributional, descriptions of the SGD dynamics that hold in non-convex situations.

In the vein of continuous approximation to stochastic algorithms, a related body of work is stochastic approximation theory (Kushner & Yin, 2003; Ljung et al., 2012), which establish ODEs as almost sure limits of trajectories of stochastic algorithms. In contrast, we obtain SDEs that are weak limits that approximate not individual sample paths, but their distributions. Other deterministic continuous time approximation methods include Su et al. (2014); Krichene et al. (2015); Wibisono et al. (2016).

Related work in SDE approximations of the SGD are Mandt et al. (2015; 2016), where the authors derived the first order SME heuristically. In contrast, we establish a rigorous statement for this type of approximations (Thm. 1). Moreover, we use asymptotic analysis and control theory to translate understanding into practical algorithms. Outside of the machine learning literature, similar modified equation methods also appear in numerical analysis of SDEs (Zygalakis, 2011) and quantifying uncertainties in ODEs (Conrad et al., 2015).

The second half of our work deals with practical problems of adaptive selection of the learning rate and momentum parameter. There is abundant literature on learning rate adjustments, including annealing schedules (Robbins & Monro, 1951; Moulines, 2011; Xu, 2011; Shamir & Zhang, 2013), adaptive per-element adjustments (Duchi et al., 2011; Zeiler, 2012; Tieleman & Hinton, 2012; Kingma & Ba, 2015) and meta-learning (Andrychowicz et al., 2016). Our approach differs in that optimal control theory provides a natural, non-black-box framework for developing dynamic feed-back adjustments, allowing us to obtain adaptive algorithms that are truly robust to changing model settings. Our learning rate adjustment policy is similar to Schaul et al. (2013); Schaul & LeCun (2013) based on one-step optimization, although we arrive at it from control theory. Our method may also be easier to implement because it does not require estimating diagonal Hessians via back-propagation. There is less literature on momentum parameter selection. A heuristic annealing schedule (referred to as MSGD-A earlier) is suggested in Sutskever et al. (2013), based on the original work of Nesterov (1983). The choice of momentum parameter in deterministic problems is discussed in Qian (1999); Nesterov (2013). To the best of our knowledge, a systematic stochastic treatment of adaptive momentum parameter selection for MSGD has not be considered before.

6. Conclusion and Outlook

Our main contribution is twofold. First, we propose the SME as a unified framework for quantifying the dynamics of SGD and its variants, beyond the classical convex regime. Tools from stochastic calculus and asymptotic analysis provide precise dynamical description of these algorithms, which help us understand important phenomena, such as descent-fluctuation transitions and the nature of acceleration schemes. Second, we use control theory as a natural framework to derive adaptive adjustment policies for the learning rate and momentum parameter. This translates to robust algorithms that requires little tuning across multiple datasets and model choices.

An interesting direction of future work is extending the SME framework to develop adaptive adjustment schemes for other hyper-parameters in SGD variants, such as Polyak-Ruppert Averaging (Polyak & Juditsky, 1992), SVRG (Johnson & Zhang, 2013) and elastic averaging SGD (Zhang et al., 2015). More generally, the SME framework may be a promising methodology for the analysis and design of stochastic gradient algorithms and beyond.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. We are also grateful for the many discussions with Dr Sixin Zhang. This work is supported in part by Major Program of NNSFC under grant 91130005, DOE DE-SC0009248, and ONR N00014-13-1-0338.

References

Andrychowicz, Marcin, Denil, Misha, Gomez, Sergio, Hoffman, Matthew W, Pfau, David, Schaul, Tom, and de Freitas, Nando. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pp. 3981–3989, 2016.

Bach, Francis and Moulines, Eric. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In *Advances in Neural Information Processing Systems*, pp. 773–781, 2013.

Bellman, Richard. Dynamic programming and Lagrange multipliers. *Proceedings of the National Academy of Sciences*, 42(10):767–769, 1956.

Conrad, Patrick R, Girolami, Mark, Särkkä, Simo, Stuart, Andrew, and Zygalakis, Konstantinos. Probability measures for numerical solutions of differential equations. *arXiv preprint arXiv:1506.04592*, 2015.

Daly, Bart J. The stability properties of a coupled pair of non-linear partial difference equations. *Mathematics of Computation*, 17(84):346–360, 1963.

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

Freidlin, Mark I, Szücs, Joseph, and Wentzell, Alexander D. *Random perturbations of dynamical systems*, volume 260. Springer Science & Business Media, 2012.

Hirt, CW. Heuristic stability theory for finite-difference equations. *Journal of Computational Physics*, 2(4):339–355, 1968.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *ICLR*, 2015.

Kloeden, P. E. and Platen, E. *Numerical Solution of Stochastic Differential Equations*. Springer, New York, corrected edition, June 2011.

Krichene, Walid, Bayen, Alexandre, and Bartlett, Peter L. Accelerated mirror descent in continuous and discrete time. In *Advances in neural information processing systems*, pp. 2845–2853, 2015.

Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. 2009.

Kushner, Harold and Yin, G George. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

LeCun, Yann, Cortes, Corinna, and Burges, Christopher JC. The mnist dataset of handwritten digits. *URL http://yann. lecun. com/exdb/mnist*, 1998.

Ljung, Lennart, Pflug, Georg Ch, and Walk, Harro. *Stochastic approximation and optimization of random systems*, volume 17. Birkhäuser, 2012.

Mandt, Stephan, Hoffman, Matthew D, and Blei, David M. Continuous-time limit of stochastic gradient descent revisited. In *NIPS-2015*, 2015.

Mandt, Stephan, Hoffman, Matthew D, and Blei, David M. A variational analysis of stochastic gradient algorithms. *arXiv preprint arXiv:1602.02666*, 2016.

Milstein, GN. *Numerical integration of stochastic differential equations*, volume 313. Springer Science & Business Media, 1995.

Moulines, Eric and, Francis R. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pp. 451–459, 2011.

Needell, Deanna, Ward, Rachel, and Srebro, Nati. Stochastic gradient descent, weighted sampling, and the randomized algorithm. In *Advances in Neural Information Processing Systems*, pp. 1017–1025, 2014.

Nesterov, Yurii. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pp. 372–376, 1983.

Nesterov, Yurii. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Noh, WF and Protter, MH. Difference methods and the equations of hydrodynamics. Technical report, California. Univ., Livermore. Lawrence Radiation Lab., 1960.

Pardalos, Panos M and Yatsenko, Vitaliy A. *Optimization and Control of Bilinear Systems: Theory, Algorithms, and Applications*, volume 11. Springer Science & Business Media, 2010.

Polyak, Boris T. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Polyak, Boris T and Juditsky, Anatoli B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Qian, Ning. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

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

Schaul, Tom and LeCun, Yann. Adaptive learning rates and parallelization for stochastic, sparse, non-smooth gradients. *arXiv preprint arXiv:1301.3764*, 2013.

Schaul, Tom, Zhang, Sixin, and LeCun, Yann. No more pesky learning rates. In *ICML (3)*, volume 28, pp. 343–351, 2013.

Shalev-Shwartz, Shai and Zhang, Tong. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, pp. 1–41, 2014.

Shamir, Ohad and Zhang, Tong. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *ICML (1)*, pp. 71–79, 2013.

Su, Weijie, Boyd, Stephen, and Candes, Emmanuel. A differential equation for modeling Nesterovs accelerated gradient method: theory and insights. In *Advances in Neural Information Processing Systems*, pp. 2510–2518, 2014.

Sutskever, Ilya, Martens, James, Dahl, George, and Hinton, Geoffrey. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pp. 1139–1147, 2013.

Tieleman, T. and Hinton, G. Lecture 6.5 - RMSProp. Technical report, 2012.

Uhlenbeck, George E and Ornstein, Leonard S. On the theory of the Brownian motion. *Physical review*, 36(5): 823, 1930.

Warming, RF and Hyett, BJ. The modified equation approach to the stability and accuracy analysis of finite-difference methods. *Journal of computational physics*, 14(2):159–179, 1974.

Wibisono, Andre, Wilson, Ashia C., and Jordan, Michael I. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016. doi: 10.1073/pnas.1614734113.

Xiao, Lin and Zhang, Tong. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Xu, Wei. Towards optimal one pass large scale learning with averaged stochastic gradient descent. *arXiv preprint arXiv:1107.2490*, 2011.

Zeiler, Matthew D. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Zhang, Sixin, Choromanska, Anna E, and LeCun, Yann. Deep learning with elastic averaging SGD. In *Advances in Neural Information Processing Systems*, pp. 685–693, 2015.

Zygalakis, KC. On the existence and the applications of modified equations for stochastic differential equations. *SIAM Journal on Scientific Computing*, 33(1):102–130, 2011.