Dissecting Adam: The Sign, Magnitude and Variance of Stochastic Gradients

Lukas Balles [1] **Philipp Hennig** [1]

Abstract

The ADAM optimizer is exceedingly popular in the deep learning community. Often it works very well, sometimes it doesn't. Why? We interpret ADAM as a combination of two aspects: for each weight, the update direction is determined by the *sign* of stochastic gradients, whereas the update magnitude is determined by an estimate of their *relative variance*. We disentangle these two aspects and analyze them in isolation, gaining insight into the mechanisms underlying ADAM. This analysis also extends recent results on adverse effects of ADAM on generalization, isolating the sign aspect as the problematic one. Transferring the variance adaptation to SGD gives rise to a novel method, completing the practitioner's toolbox for problems where ADAM fails.

1. Introduction

Many prominent machine learning models pose empirical risk minimization problems with objectives of the form

$$\mathcal{L}(\theta) = \frac{1}{M}\sum_{k=1}^{M} \ell(\theta; x_k), \tag{1}$$

$$\nabla\mathcal{L}(\theta) = \frac{1}{M}\sum_{k=1}^{M} \nabla\ell(\theta; x_k), \tag{2}$$

where $\theta \in \mathbb{R}^d$ is a vector of parameters, $\{x_1, \dots, x_M\}$ is a training set, and $\ell(\theta; x)$ is a loss quantifying the performance of parameters θ on example x. Computing the exact gradient in each step of an iterative optimization algorithm becomes inefficient for large M. Instead, we sample a mini-batch $\mathcal{B} \subset \{1, \dots, M\}$ of size $|\mathcal{B}| \ll M$ with data points drawn uniformly and independently from the training set and compute an approximate *stochastic gradient*

$$g(\theta) = \frac{1}{|\mathcal{B}|}\sum_{k\in\mathcal{B}} \nabla\ell(\theta; x_k), \tag{3}$$

which is a random variable with $\mathbf{E}[g(\theta)] = \nabla\mathcal{L}(\theta)$. An important quantity for this paper will be the (element-wise) variances of the stochastic gradient, which we denote by $\sigma_i^2(\theta) := \mathbf{var}[g(\theta)_i]$.

Widely-used stochastic optimization algorithms are stochastic gradient descent (SGD, Robbins & Monro, 1951) and its momentum variants (Polyak, 1964; Nesterov, 1983). A number of methods popular in deep learning choose per-element update magnitudes based on past gradient observations. Among these are ADAGRAD (Duchi et al., 2011), RMSPROP (Tieleman & Hinton, 2012), ADADELTA (Zeiler, 2012), and ADAM (Kingma & Ba, 2015).

Notation: In the following, we occasionally drop θ, writing g instead of $g(\theta)$, et cetera. We use shorthands like $\nabla\mathcal{L}_t$, g_t for sequences θ_t and double indices where needed, e.g., $g_{t,i} = g(\theta_t)_i$, $\sigma_{t,i}^2 = \sigma_i^2(\theta_t)$. Divisions, squares and square-roots on vectors are to be understood *element-wise*. To avoid confusion with inner products, we explicitly denote element-wise multiplication of vectors by $\odot$.

1.1. A New Perspective on Adam

We start out from a reinterpretation of the widely-used ADAM optimizer,[2] which maintains moving averages of stochastic gradients and their element-wise square,

$$\tilde{m}_t = \beta_1\tilde{m}_{t-1} + (1-\beta_1)g_t, \quad m_t = \frac{\tilde{m}_t}{1-\beta_1^{t+1}}, \tag{4}$$

$$\tilde{v}_t = \beta_2\tilde{v}_{t-1} + (1-\beta_2)g_t^2, \quad v_t = \frac{\tilde{v}_t}{1-\beta_2^{t+1}}, \tag{5}$$

with $\beta_1, \beta_2 \in (0, 1)$ and updates

$$\theta_{t+1} = \theta_t - \alpha\frac{m_t}{\sqrt{v_t}+\varepsilon} \tag{6}$$

with a small constant $\varepsilon > 0$ preventing division by zero. Ignoring ε and assuming $|m_{t,i}| > 0$ for the moment, we can rewrite the update direction as

$$\frac{m_t}{\sqrt{v_t}} = \frac{\operatorname{sign}(m_t)}{\sqrt{\frac{v_t}{m_t^2}}} = \sqrt{\frac{1}{1+\frac{v_t-m_t^2}{m_t^2}}} \odot \operatorname{sign}(m_t), \tag{7}$$

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany.. Correspondence to: Lukas Balles <lballes@tue.mpg.de>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

[2]Some of our considerations naturally extend to ADAM's relatives RMSPROP and ADADELTA, but we restrict our attention to ADAM to keep the presentation concise.

where the sign is to be understood element-wise. Assuming that m_t and v_t approximate the first and second moment of the stochastic gradient—a notion that we will discuss further in §4.1—$(v_t - m_t^2)$ can be seen as an estimate of the stochastic gradient variances. The use of the *non-central* second moment effectively cancels out the magnitude of m_t; it *only* appears in the ratio $(v_t - m_t^2)/m_t^2$. Hence, ADAM can be interpreted as a combination of two aspects:

- The update *direction* for the i-th coordinate is given by the *sign* of $m_{t,i}$.
- The update *magnitude* for the i-th coordinate is solely determined by the global step size α and the factor

$$\gamma_{t,i} := \sqrt{\frac{1}{1+\hat{\eta}_{t,i}^2}}, \tag{8}$$

where $\hat{\eta}_{t,i}$ is an estimate of the *relative variance*,

$$\hat{\eta}_{t,i}^2 := \frac{v_{t,i} - m_{t,i}^2}{m_{t,i}^2} \approx \frac{\sigma_{t,i}^2}{\nabla \mathcal{L}_{t,i}^2} =: \eta_{t,i}^2. \tag{9}$$

We will refer to the second aspect as *variance adaptation*. The variance adaptation factors shorten the update in directions of high relative variance, adapting for varying reliability of the stochastic gradient in different coordinates.

The above interpretation of ADAM's update rule has to be viewed in contrast to existing ones. A motivation given by Kingma & Ba (2015) is that v_t is a diagonal approximation to the empirical Fisher information matrix (FIM), making ADAM an approximation to natural gradient descent (Amari, 1998). Apart from fundamental reservations towards the *empirical* Fisher and the quality of *diagonal* approximations (Martens, 2014, §11), this view is problematic because the FIM, if anything, is approximated by v_t, whereas ADAM adapts with the square-root $\sqrt{v_t}$.

Another possible motivation (which is not found in peer-reviewed publications but circulates the community as "conventional wisdom") is that ADAM performs an approximate *whitening* of stochastic gradients. However, this view hinges on the fact that ADAM divides by the square-root of the *non-central* second moment, not by the standard deviation.

1.2. Overview

Both aspects of ADAM—taking the sign and variance adaptation—are briefly mentioned in Kingma & Ba (2015), who note that "[t]he effective stepsize [...] is also invariant to the scale of the gradients" and refer to $m_t/\sqrt{v_t}$ as a "signal-to-noise ratio". The purpose of this work is to disentangle these two aspects in order to discuss and analyze them in isolation.

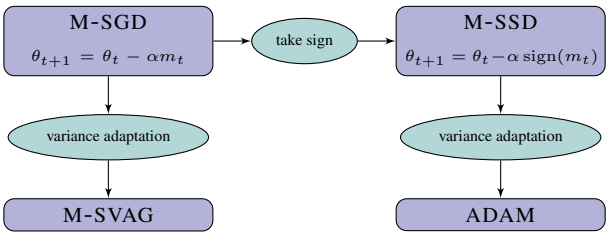

Figure 1. The methods under consideration in this paper. "M-" refers to the use of m_t in place of g_t, which we colloquially refer to as the *momentum variant*. M-SVAG will be derived below.

This perspective naturally suggests two alternative methods by incorporating one of the aspects while excluding the other. Taking the sign of a stochastic gradient without any further modification gives rise to *Stochastic Sign Descent* (SSD). On the other hand, *Stochastic Variance-Adapted Gradient* (SVAG), to be derived in §3.2, applies variance adaptation directly to the stochastic gradient instead of its sign. Together with ADAM, the momentum variants of SGD, SSD, and SVAG constitute the four possible recombinations of the sign aspect and the variance adaptation, see Fig. 1.

We proceed as follows: Section 2 discusses the sign aspect. In a simplified setting we investigate under which circumstances the sign of a stochastic gradient is a better update direction than the stochastic gradient itself. Section 3 presents a principled derivation of element-wise variance adaptation factors. Subsequently, we discuss the practical implementation of variance-adapted methods (Section 4). Section 5 draws a connection to recent work on ADAM's effect on generalization. Finally, Section 6 presents experimental results.

1.3. Related Work

Sign-based optimization algorithms have received some attention in the past. RPROP (Riedmiller & Braun, 1993) is based on gradient signs and adapts per-element update magnitudes based on observed sign changes. Seide et al. (2014) empirically investigate the use of stochastic gradient signs in a distributed setting with the goal of reducing communication cost. Karimi et al. (2016) prove convergence results for sign-based methods in the *non-stochastic* case.

Variance-based update directions have been proposed before, e.g., by Schaul et al. (2013), where the variance appears together with curvature estimates in a diagonal preconditioner for SGD. Their variance-dependent terms resemble the variance adaptation factors we will derive in Section 3. The corresponding parts of our work complement that of Schaul et al. (2013) in various ways. Most notably, we provide a principled motivation for variance adaptation that is independent of the update direction and use that to extend the variance adaptation to the momentum case.

A somewhat related line of research aims to obtain *reduced-variance* gradient estimates (e.g., Johnson & Zhang, 2013; Defazio et al., 2014). This is largely orthogonal to our notion of variance adaptation, which alters the search direction to mitigate adverse effects of the (remaining) variance.

1.4. The Sign of a Stochastic Gradient

For later use, we briefly establish some facts about the sign[3] of a stochastic gradient, $s = \operatorname{sign}(g)$. The distribution of the binary random variable s_i is fully characterized by the *success probability* $\rho_i := \mathbf{P}\left[s_i = \operatorname{sign}(\nabla \mathcal{L}_i)\right]$, which generally depends on the distribution of g_i. If we assume g_i to be normally distributed, which is supported by the Central Limit Theorem applied to Eq. (3), we have

$$\rho_i = \frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{|\nabla \mathcal{L}_i|}{\sqrt{2}\sigma_i}\right), \tag{10}$$

see §B.1 of the supplementary material. Note that ρ_i is uniquely determined by the relative variance of g_i.

2. Why the Sign?

Can it make sense to use the sign of a stochastic gradient as the update direction instead of the stochastic gradient itself? This question is difficult to tackle in a general setting, but we can get an intuition using the simple, yet insightful, case of stochastic quadratic problems, where we can investigate the effects of curvature properties and noise.

Model Problem (Stochastic Quadratic Problem, sQP)**.** *Consider the loss function $\ell(\theta; x) = 0.5\,(\theta - x)^T Q (\theta - x)$ with a symmetric positive definite matrix $Q \in \mathbb{R}^{d \times d}$ and "data" coming from the distribution $x \sim \mathcal{N}(x^*, \nu^2 I)$ with $\nu \in \mathbb{R}_+$. The objective $\mathcal{L}(\theta) = \mathbf{E}_x[\ell(\theta; x)]$ evaluates to*

$$\mathcal{L}(\theta) = \frac{1}{2}(\theta - x^*)^T Q (\theta - x^*) + \frac{\nu^2}{2} \operatorname{tr}(Q), \tag{11}$$

with $\nabla \mathcal{L}(\theta) = Q(\theta - x^*)$. Stochastic gradients are given by $g(\theta) = Q(\theta - x) \sim \mathcal{N}(\nabla \mathcal{L}(\theta), \nu^2 Q Q)$.

2.1. Theoretical Comparison

We compare update directions on sQPs in terms of their local expected decrease in function value from a single step. For any stochastic direction z, updating from θ to $\theta + \alpha z$ results in $\mathbf{E}[\mathcal{L}(\theta + \alpha z)] = \mathcal{L}(\theta) + \alpha \nabla \mathcal{L}(\theta)^T \mathbf{E}[z] + \frac{\alpha^2}{2} \mathbf{E}[z^T Q z]$. For this comparison of update *directions* we use the optimal step size minimizing $\mathbf{E}[\mathcal{L}(\theta + \alpha z)]$, which is easily found to be $\alpha_* = -\nabla \mathcal{L}(\theta)^T \mathbf{E}[z] / \mathbf{E}[z^T Q z]$ and yields an expected improvement of

$$\mathcal{I}(\theta) := |\mathbf{E}[\mathcal{L}(\theta + \alpha_* z)] - \mathcal{L}(\theta)| = \frac{(\nabla \mathcal{L}(\theta)^T \mathbf{E}[z])^2}{2 \mathbf{E}[z^T Q z]}. \tag{12}$$

[3] To avoid a separate zero-case, we define $\operatorname{sign}(0) = 1$ for all theoretical considerations. Note that $g_i \neq 0$ a.s. if $\mathbf{var}[g_i] > 0$.

Locally, a larger expected improvement implies a better update direction. We compute this quantity for SGD ($z = -g(\theta)$) and SSD ($z = -\operatorname{sign}(g(\theta))$) in §B.2 of the supplementary material and find

$$\mathcal{I}_{\text{SGD}}(\theta) = \frac{1}{2} \frac{(\nabla \mathcal{L}(\theta)^T \nabla \mathcal{L}(\theta))^2}{\nabla \mathcal{L}(\theta)^T Q \nabla \mathcal{L}(\theta) + \nu^2 \sum_{i=1}^d \lambda_i^3}, \tag{13}$$

$$\mathcal{I}_{\text{SSD}}(\theta) \geq \frac{1}{2} \frac{\left(\sum_{i=1}^d (2\rho_i - 1) |\nabla \mathcal{L}(\theta)_i|\right)^2}{\sum_{i=1}^d \lambda_i} p_{\text{diag}}(Q), \tag{14}$$

where the $\lambda_i \in \mathbb{R}_+$ are the eigenvalues of Q and $p_{\text{diag}}(Q) := (\sum_{i=1}^d |q_{ii}|) / (\sum_{i,j=1}^d |q_{ij}|)$ measures the percentage of diagonal mass of Q. $\mathcal{I}_{\text{SGD}}$ and $\mathcal{I}_{\text{SSD}}$ are *local* quantities, depending on θ, which makes a general and conclusive comparison impossible. However, we can draw conclusions about how properties of the sQP affect the two update directions. We make the following two observations:

Firstly, the term $p_{\text{diag}}(Q)$, which features only in $\mathcal{I}_{\text{SSD}}$, relates to the orientation of the eigenbasis of Q. If Q is diagonal, the problem is perfectly axis-aligned and we have $p_{\text{diag}}(Q) = 1$. This is the obvious best case for the intrinsically axis-aligned sign update. However, $p_{\text{diag}}(Q)$ can become as small as $1/d$ in the worst case and will on average (over random orientations) be $p_{\text{diag}}(Q) \approx 1.57/d$. (We show these properties in §B.2 of the supplementary material.) This suggests that the sign update will have difficulties with arbitrarily-rotated eigenbases and crucially relies on the problem being "close to axis-aligned".

Secondly, $\mathcal{I}_{\text{SGD}}$ contains the term $\nu^2 \sum_{i=1}^d \lambda_i^3$ in which stochastic noise and the eigenspectrum of the problem *interact*. $\mathcal{I}_{\text{SSD}}$, on the other hand, has a milder dependence on the eigenvalues of Q and there is no such interaction between noise and eigenspectrum. The noise only manifests in the element-wise success probabilities ρ_i.

In summary, we can expect the sign direction to be beneficial for noisy, ill-conditioned problems with diagonally dominant Hessians. It is unclear to what extent these properties hold for real problems, on which sign-based methods like ADAM are usually applied. Becker & LeCun (1988) empirically investigated the first property for Hessians of simple neural network training problems and found comparably high values of $p_{\text{diag}}(Q) = 0.1$ up to $p_{\text{diag}}(Q) = 0.6$. Chaudhari et al. (2017) empirically investigated the eigenspectrum in deep learning problems and found it to be very ill-conditioned with the majority of eigenvalues close to zero and a few very large ones. However, this empirical evidence is far from conclusive.

2.2. Experimental Evaluation

We verify our findings experimentally on 100-dimensional sQPs. First, we specify a diagonal matrix $\Lambda \in \mathbb{R}^{100}$ of

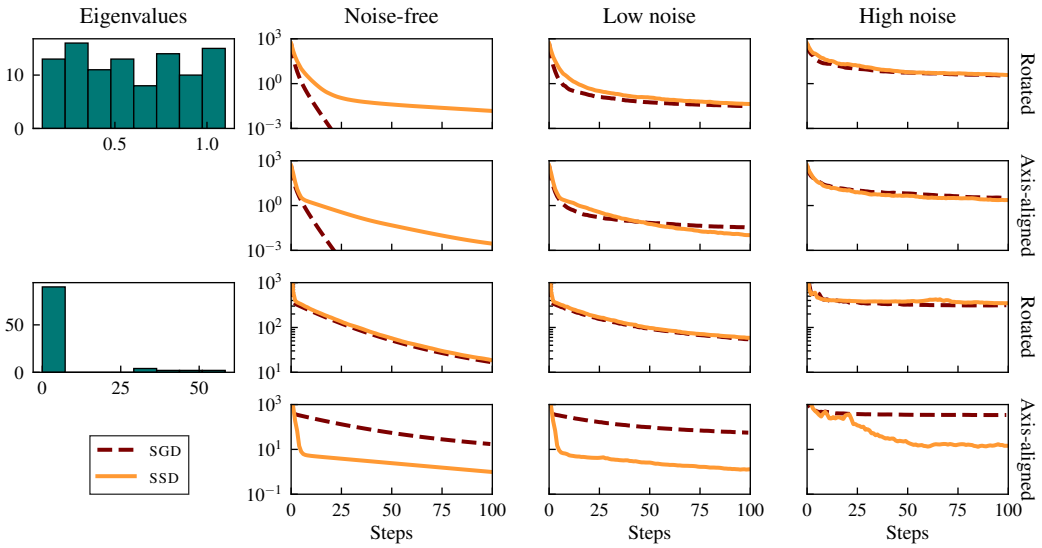

Figure 2. Performance of SGD and SSD on stochastic quadratic problems. Rows correspond to different QPs: the eigenspectrum is shown and each is used with a randomly rotated and an axis-aligned eigenbasis. Columns correspond to different noise levels. The individual panels show function value over number of steps. On the well-conditioned problem, gradient descent vastly outperforms the sign-based method in the noise-free case, but the difference is evened out when noise is added. The orientation of the eigenbasis had little effect on the comparison in the well-conditioned case. On the ill-conditioned problem, the methods perform roughly equal when the eigenbasis is randomly rotated. SSD benefits drastically from an axis-aligned eigenbasis, where it clearly outperforms SGD.

eigenvalues: (a) a mildly-conditioned problem with values drawn uniformly from $[0.1, 1.1]$ and (b) an ill-conditioned problem with a structured eigenspectrum simulating the one reported by Chaudhari et al. (2017) by uniformly drawing 90% of the values from $[0, 1]$ and 10% from $[30, 60]$. Q is then defined as (a) $Q = \Lambda$ for an axis-aligned problem and (b) $Q = R\Lambda R^T$ with a random R drawn uniformly among all rotation matrices (see Diaconis & Shahshahani, 1987). This makes four different matrices, which we consider at noise levels $\nu \in \{0, 0.1, 4.0\}$. We run SGD and SSD with their optimal local step sizes as previously derived. The results, shown in Fig. 2, confirm our theoretical findings.

3. Variance Adaptation

We now proceed to the second component of ADAM: variance-based element-wise step sizes. Considering this variance adaptation in isolation from the sign aspect naturally suggests to employ it on arbitrary update directions, for example directly on the stochastic gradient instead of its sign. A principled motivation arises from the following consideration:

Assume we want to update in a direction $p \in \mathbb{R}^d$ (or $\text{sign}(p)$), but only have access to an estimate $\hat{p}$ with $\mathbf{E}[\hat{p}] = p$. We allow element-wise factors $\gamma \in \mathbb{R}^d$ and update $\gamma \odot \hat{p}$ (or $\gamma \odot \text{sign}(\hat{p})$). One way to make "optimal" use of these factors is to choose them such as to minimize the expected distance to the desired update direction.

Lemma 1. *Let $\hat{p} \in \mathbb{R}^d$ be a random variable with $\mathbf{E}[\hat{p}] = p$ and $\mathbf{var}[p_i] = \sigma_i^2$. Then $\mathbf{E}[\|\gamma \odot \hat{p} - p\|_2^2]$ is minimized by*

$$\gamma_i = \frac{\mathbf{E}[\hat{p}_i]^2}{\mathbf{E}[\hat{p}_i^2]} = \frac{p_i^2}{p_i^2 + \sigma_i^2} = \frac{1}{1 + \sigma_i^2 / p_i^2} \tag{15}$$

and $\mathbf{E}[\|\gamma \odot \text{sign}(\hat{p}) - \text{sign}(p)\|_2^2]$ is minimized by

$$\gamma_i = (2\rho_i - 1), \tag{16}$$

where $\rho_i := \mathbf{P}[\text{sign}(\hat{p}_i) = \text{sign}(p_i)]$. *(Proof in §B.3)*

3.1. ADAM as Variance-Adapted Sign Descent

According to Lemma 1, the optimal variance adaptation factors for the sign of a stochastic gradient are $\gamma_i = 2\rho_i - 1$, where $\rho_i = \mathbf{P}[\text{sign}(g_i) = \text{sign}(\nabla \mathcal{L}_i)]$. Appealing to intuition, this means that γ_i is proportional to the success probability with a maximum of 1 when we are certain about the sign of the gradient ($\rho_i = 1$) and a minimum of 0 in the absence of information ($\rho_i = 0.5$).

Recall from Eq. (10) that, under the Gaussian assumption, the success probabilities are $2\rho_i - 1 = \text{erf}[(\sqrt{2}\eta_i)^{-1}]$. Figure 3 shows that this term is closely approximated by $(1+\eta_i^2)^{-1/2}$, the variance adaptation terms of ADAM. Hence, ADAM can be regarded as an approximate realization of this optimal variance adaptation scheme. This comes with the caveat that ADAM applies these factors to $\text{sign}(m_t)$ instead of $\text{sign}(g_t)$. Variance adaptation for m_t will be discussed further in §4.3 and in the supplements §C.2.

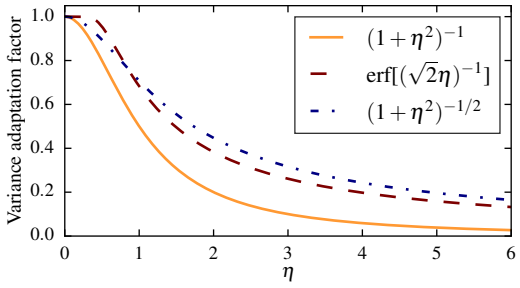

Figure 3. Variance adaptation factors as functions of the relative standard deviation η. The optimal factor for the sign of a (Gaussian) stochastic gradient is $\mathrm{erf}[(\sqrt{2}\eta)^{-1}]$, which is closely approximated by $(1+\eta^2)^{-1/2}$, the factor implicitly employed by ADAM. $(1+\eta^2)^{-1}$ is the optimal factor for a stochastic gradient.

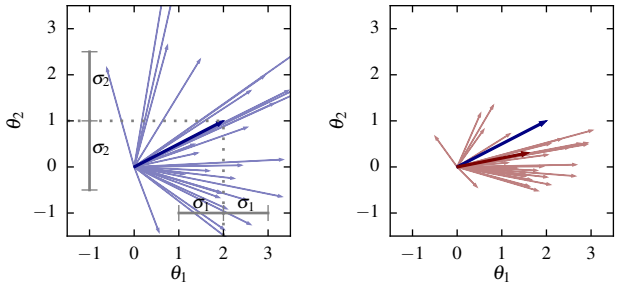

Figure 4. Conceptual sketch of variance-adapted stochastic gradients. The left panel shows the true gradient $\nabla\mathcal{L} = (2, 1)$ and stochastic gradients scattered around it with $(\sigma_1, \sigma_2) = (1, 1.5)$. In the right panel, we scale the i-th coordinate by $(1+\eta_i^2)^{-1}$. In this example, the θ_2-coordinate has much higher relative variance ($\eta_2^2 = 2.25$) than the θ_1-coordinate ($\eta_1^2 = 0.25$) and is thus shortened. This reduces the variance of the update direction at the expense of biasing it away from the true gradient in expectation.

3.2. Stochastic Variance-Adapted Gradient (SVAG)

Applying Eq. (15) to $\hat{p} = g$, the optimal variance adaptation factors for a stochastic gradient are found to be

$$\gamma_i^g = \frac{\nabla\mathcal{L}_i^2}{\nabla\mathcal{L}_i^2 + \sigma_i^2} = \frac{1}{1 + \sigma_i^2/\nabla\mathcal{L}_i^2} = \frac{1}{1+\eta_i^2}. \qquad (17)$$

A term of this form also appears, together with diagonal curvature estimates, in Schaul et al. (2013). We refer to the method updating along $\gamma^g \odot g$ as *Stochastic Variance-Adapted Gradient* (SVAG). To support intuition, Fig. 4 shows a conceptual sketch of this variance adaptation scheme.

Variance adaptation of this form guarantees convergence *without* manually decreasing the global step size. We recover the $\mathcal{O}(1/t)$ rate of SGD for smooth, strongly convex functions. We emphasize that this result considers an *idealized* version of SVAG with exact γ_i^g. It should be considered as a motivation for this variance adaptation strategy, not a statement about its performance with estimated variance adaptation factors.

Theorem 1. *Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be μ-strongly convex and L-smooth. We update $\theta_{t+1} = \theta_t - \alpha(\gamma_t \odot g_t)$, with stochastic gradients $\mathbf{E}[g_t|\theta_t] = \nabla f_t$, $\mathbf{var}[g_{t,i}|\theta_t] = \sigma_{t,i}^2$, variance adaptation factors $\gamma_{t,i} = \nabla f_{t,i}^2/(\nabla f_{t,i}^2 + \sigma_{t,i}^2)$, and a global step size $\alpha = 1/L$. Assume that there are constants $c_v, M_v > 0$ such that $\sum_{i=1}^d \sigma_{t,i}^2 \le c_v\|\nabla f_t\|^2 + M_v$. Then*

$$\mathbf{E}[f(\theta_t) - f_*] \in \mathcal{O}\left(\frac{1}{t}\right), \qquad (18)$$

where f_* is the minimum value of f. *(Proof in §B.4)*

The assumption $\sum_{i=1}^d \sigma_{t,i}^2 \le c_v\|\nabla f_t\|^2 + M_v$ is a mild restriction on the variances, allowing them to be non-zero everywhere and to grow quadratically in the gradient norm.

4. Practical Implementation of M-SVAG

Section 3 has introduced the general idea of variance adaptation; we now discuss its practical implementation. For the sake of a concise presentation, we focus on one particular variance-adapted method, M-SVAG, which applies variance adaptation to the update direction m_t. This method is of particular interest due to its relationship to ADAM outlined in Figure 1. Many of the following considerations correspondingly apply to other variance-adapted methods, e.g., SVAG and variants of ADAM, some of which are discussed and evaluated in the supplementary material (§C).

4.1. Estimating Gradient Variance

In practice, the optimal variance adaptation factors are unknown and have to be estimated. A key ingredient is an estimate of the stochastic gradient variance. We have argued in the introduction that ADAM obtains such an estimate from moving averages, $\sigma_{t,i}^2 \approx v_{t,i} - m_{t,i}^2$. The underlying assumption is that the distribution of stochastic gradients is approximately constant over the effective time horizon of the exponential moving average, making m_t and v_t estimates of the first and second moment of g_t, respectively:

Assumption 1. *At step t, assume*

$$\mathbf{E}[m_{t,i}] \approx \nabla\mathcal{L}_{t,i}, \quad \mathbf{E}[v_{t,i}] \approx \nabla\mathcal{L}_{t,i}^2 + \sigma_{t,i}^2. \qquad (19)$$

While this can only ever hold approximately, Assumption 1 is the tool we need to obtain gradient variance estimates from past gradient observations. It will be more realistic in the case of high noise and small step size, where the variation between successive stochastic gradients is dominated by stochasticity rather than change in the true gradient.

We make two modifications to ADAM's variance estimate. First, we will use the same moving average constant $\beta_1 =$

$\beta_2 = \beta$ for m_t and v_t. This constant should define the effective range for which we implicitly assume the stochastic gradients to come from the same distribution, making different constants for the first and second moment implausible.

Secondly, we adapt for a systematic bias in the variance estimate. As we show in §B.5, under Assumption 1,

$$\mathbf{E}[m_{t,i}^2] \approx \nabla\mathcal{L}_{t,i}^2 + \rho(\beta,t)\sigma_{t,i}^2, \tag{20}$$

$$\rho(\beta,t) := \frac{(1-\beta)(1+\beta^{t+1})}{(1+\beta)(1-\beta^{t+1})}, \tag{21}$$

and consequently $\mathbf{E}[v_{t,i} - m_{t,i}^2] \approx (1-\rho(\beta,t))\,\sigma_{t,i}^2$. We correct for this bias and use the variance estimate

$$\hat{s}_t := \frac{1}{1-\rho(\beta,t)}(v_t - m_t^2). \tag{22}$$

Mini-Batch Gradient Variance Estimates: An alternative variance estimate can be computed locally "within" a single mini-batch, see §D of the supplements. We have experimented with both estimators and found the resulting methods to have similar performance. For the main paper, we stick to the moving average variant for its ease of implementation and direct correspondence with ADAM. We present experiments with the mini-batch variant in the supplementary material. These demonstrate the merit of variance adaptation *irrespective* of how the variance is estimated.

4.2. Estimating the Variance Adaptation Factors

The gradient variance itself is not of primary interest; we have to estimate the variance adaptation factors, given by Eq. (17) in the case of SVAG. We propose to use the estimate

$$\hat{\gamma}_t^g = \frac{1}{1+\hat{s}_t/m_t^2} = \frac{m_t^2}{m_t^2+\hat{s}_t}. \tag{23}$$

While $\hat{\gamma}_t^g$ is an intuitive quantity, it is *not* an unbiased estimate of the exact variance adaptation factors as defined in Eq. (17). To our knowledge, unbiased estimation of the exact factors is intractable. We have experimented with several partial bias correction terms but found them to have destabilizing effects.

4.3. Incorporating Momentum

So far, we have considered variance adaptation for the update direction g_t. In practice, we may want to update in the direction of m_t to incorporate momentum.[4] According to Lemma 1, the variance adaptation factors should then be determined by the relative of variance of m_t.

Once more adopting Assumption 1, we have $\mathbf{E}[m_t] \approx \nabla\mathcal{L}_t$ and $\mathbf{var}[m_{t,i}] \approx \rho(\beta,t)\sigma_{t,i}^2$, the latter being due to Eq. (20). Hence, the relative variance of m_t is $\rho(\beta,t)$ times that of g_t, such that the optimal variance adaptation factors for the update direction m_t according to Lemma 1 are

$$\gamma_{t,i}^m = \frac{1}{1+\rho(\beta,t)\sigma_{t,i}^2/\nabla\mathcal{L}_{t,i}^2}. \tag{24}$$

We use the following estimate thereof:

$$\hat{\gamma}_t^m = \frac{1}{1+\rho(\beta,t)\,\hat{s}_t/m_t^2} = \frac{m_t^2}{m_t^2+\rho(\beta,t)\,\hat{s}_t}. \tag{25}$$

Note that m_t now serves a double purpose: It determines the base update direction and, at the same time, is used to obtain an estimate of the gradient variance.

4.4. Details

Note that Eq. (22) is ill-defined for $t=0$, since $\rho(\beta,0)=0$. We use $\hat{s}_0 = 0$ for the first iteration, making the initial step of M-SVAG coincide with an SGD-step. One final detail concerns a possible division by zero in Eq. (25). Unlike ADAM, we do not add a constant offset ε in the denominator. A division by zero only occurs when $m_{t,i} = v_{t,i} = 0$; we check for this case and perform no update, since $m_{t,i} = 0$.

This completes the description of our implementation of M-SVAG. Alg. 1 provides pseudo-code (ignoring the details discussed in §4.4 for readability).

Algorithm 1 M-SVAG

Input: $\theta_0 \in \mathbb{R}^d$, $\alpha > 0$, $\beta \in [0,1]$, $T \in \mathbb{N}$
Initialize $\theta \leftarrow \theta_0$, $\tilde{m} \leftarrow 0$, $\tilde{v} \leftarrow 0$
for $t = 0, \dots, T-1$ **do**
  $\tilde{m} \leftarrow \beta\tilde{m} + (1-\beta)g(\theta)$, $\quad \tilde{v} \leftarrow \beta\tilde{v} + (1-\beta)g(\theta)^2$
  $m \leftarrow (1-\beta^{t+1})^{-1}\tilde{m}$, $\quad v \leftarrow (1-\beta^{t+1})^{-1}\tilde{v}$
  $s \leftarrow (1-\rho(\beta,t))^{-1}(v-m^2)$
  $\gamma \leftarrow m^2/(m^2+\rho(\beta,t)s)$
  $\theta \leftarrow \theta - \alpha(\gamma \odot m)$
end for

Note: M-SVAG exposes two hyperparameters, α and β.

5. Connection to Generalization

Of late, the question of the effect of the optimization algorithm on *generalization* has received increased attention. Especially in deep learning, different optimizers might find solutions with varying generalization performance. Recently, Wilson et al. (2017) have argued that "adaptive methods" (referring to ADAGRAD, RMSPROP, and ADAM) have adverse effects on generalization compared to "non-adaptive

[4] Our use of the term *momentum* is somewhat colloquial. To highlight the relationship with ADAM (Fig. 1), we have defined M-SGD as the method using the update direction m_t, which is a rescaled version of SGD with momentum. M-SVAG applies variance adaptation to m_t. This is not to be confused with the application of momentum acceleration (Polyak, 1964; Nesterov, 1983) *on top* of a SVAG update.

methods" (gradient descent, SGD, and their momentum variants). In addition to an extensive empirical validation of that claim, the authors make a theoretical argument using a binary least-squares classification problem,

$$R(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(x_i^T\theta - y_i)^2 = \frac{1}{2n}\|X\theta - y\|^2, \quad (26)$$

with n data points $(x_i, y_i) \in \mathbb{R}^d \times \{\pm 1\}$, stacked in a matrix $X \in \mathbb{R}^{n\times d}$ and a label vector $y \in \{\pm 1\}^n$. For this problem class, the non-adaptive methods provably converge to the max-margin solution, which we expect to have favorable generalization properties. In contrast to that, Wilson et al. (2017) show that—for *some* instances of this problem class—the adaptive methods converge to solutions that generalize arbitrarily bad to unseen data. The authors construct such problematic instances using the following Lemma.

Lemma 2 (Lemma 3.1 in Wilson et al. (2017))**.** *Suppose* $[X^Ty]_i \neq 0$ *for* $i = 1, \dots, d$*, and there exists* $c \in \mathbb{R}$ *such that* $X\,\mathrm{sign}(X^Ty) = cy$*. Then, when initialized at* $\theta_0 = 0$*, the iterates generated by full-batch* ADAGRAD*,* ADAM*, and* RMSPROP *on the objective* (26) *satisfy* $\theta_t \propto \mathrm{sign}(X^Ty)$*.*

Intriguingly, as we show in §B.6 of the supplementary material, this statement easily extends to sign descent, i.e., the method updating $\theta_{t+1} = \theta_t - \alpha\,\mathrm{sign}(\nabla R(\theta_t))$.

Lemma 3. *Under the assumptions of Lemma 2, the iterates generated by sign descent satisfy* $\theta_t \propto \mathrm{sign}(X^Ty)$*.*

On the other hand, this does *not* extend to M-SVAG, an *adaptive* method by any standard. As noted before, the first step of M-SVAG coincides with a gradient descent step. The iterates generated by M-SVAG will, thus, not generally be proportional to $\mathrm{sign}(X^Ty)$. While this does by no means imply that it converges to the max-margin solution or has otherwise favorable generalization properties, the construction of Wilson et al. (2017) does *not* apply to M-SVAG.

This suggests that it is the sign that impedes generalization in the examples constructed by Wilson et al. (2017), rather than the element-wise adaptivity as such. Our experiments substantiate this suspicion. The fact that all currently popular adaptive methods are also sign-based has led to a conflation of these two aspects. The main motivation for this work was to disentangle them.

6. Experiments

We experimentally compare M-SVAG and ADAM to their non-variance-adapted counterparts M-SGD and M-SSD (Alg. 2). Since these are the four possible recombinations of the sign and the variance adaptation (Fig. 1), this comparison allows us to separate the effects of the two aspects.

Algorithm 2 M-SGD and M-SSD

Input: $\theta_0 \in \mathbb{R}^d$, $\alpha > 0$, $\beta \in [0, 1]$, $T \in \mathbb{N}$
Initialize $\theta \leftarrow \theta_0$, $\tilde{m} \leftarrow 0$
for $t = 0, \dots, T-1$ **do**
  $\tilde{m} \leftarrow \beta\tilde{m} + (1-\beta)g(\theta)$
  $m \leftarrow (1-\beta^{t+1})^{-1}\tilde{m}$
  $\theta \leftarrow \theta - \alpha m$ $\theta \leftarrow \theta - \alpha\,\mathrm{sign}(\tilde{m})$
end for

6.1. Experimental Set-Up

We evaluated the four methods on the following problems:

- P1 A vanilla convolutional neural network (CNN) with two convolutional and two fully-connected layers on the Fashion-MNIST data set (Xiao et al., 2017).
- P2 A vanilla CNN with three convolutional and three fully-connected layers on CIFAR-10 (Krizhevsky, 2009).
- P3 The wide residual network WRN-40-4 architecture of Zagoruyko & Komodakis (2016) on CIFAR-100.
- P4 A two-layer LSTM (Hochreiter & Schmidhuber, 1997) for character-level language modelling on Tolstoy's *War and Peace*.

A detailed description of all network architectures has been moved to §A of the supplementary material.

For all experiments, we used $\beta = 0.9$ for M-SGD, M-SSD and M-SVAG and default parameters ($\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 10^{-8}$) for ADAM. The global step size α was tuned for each method individually by first finding the maximal stable step size by trial and error, then searching downwards. We selected the one that yielded maximal test accuracy within a fixed number of training steps; a scenario close to an actual application of the methods by a practitioner. (Loss and accuracy have been evaluated at a fixed interval on the full test set as well as on an equally-sized portion of the training set). Experiments with the best step size have been replicated ten times with different random seeds. While (P1) and (P2) were trained with constant α, we used a decrease schedule for (P3) and (P4), which was fixed in advance for all methods. Full details can be found in §A of the supplements.

6.2. Results

Fig. 5 shows results. We make four main observations.

1) The sign aspect dominates With the exception of (P4), the performance of the four methods distinctly clusters into sign-based and non-sign-based methods. Of the two components of ADAM identified in §1.1, the sign aspect seems

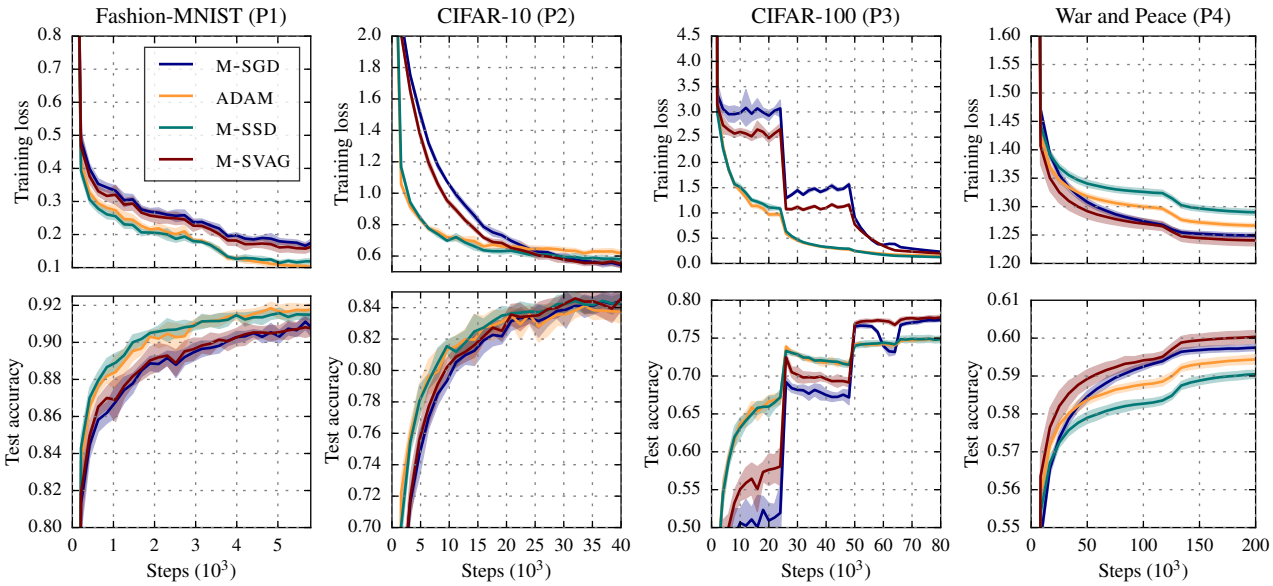

Figure 5. Experimental results on the four test problems. Plots display training loss and test accuracy over the number of steps. Curves for the different optimization methods are color-coded. The shaded area spans one standard deviation, obtained from ten replications with different random seeds.

to be by far the dominant one, accounting for most of the difference between ADAM and M-SGD. ADAM and M-SSD display surprisingly similar performance; an observation that might inform practitioners' choice of algorithm, especially for very high-dimensional problems, where ADAM's additional memory requirements are an issue.

2) The usefulness of the sign is problem-dependent Considering only training loss, the two sign-based methods clearly outperform the two non-sign-based methods on problems (P1) and (P3). On (P2), ADAM and M-SSD make rapid initial progress, but later plateau and are undercut by M-SGD and M-SVAG. On the language modelling task (P4) the non-sign-based methods show superior performance. Relating to our analysis in Section 2, this shows that the usefulness of sign-based methods depends on the particular problem at hand.

3) Variance adaptation helps In all experiments, the variance-adapted variants perform at least as good as, and often better than, their "base algorithms". The magnitude of the effect varies. For example, ADAM and M-SSD have identical performance on (P3), but M-SVAG significantly outperforms M-SGD on (P3) as well as (P4).

4) Generalization effects are caused by the sign The CIFAR-100 example (P3) displays similar effects as reported by Wilson et al. (2017): ADAM vastly outperforms M-SGD in training loss, but has significantly worse test performance. Observe that M-SSD behaves almost identical to ADAM in both train and test and, thus, displays the same generalization-harming effects. M-SVAG, on the other hand, improves upon M-SGD and, in particular, does not display any adverse effects on generalization. This corroborates the suspicion raised in §5 that the generalization-harming effects of ADAM are caused by the sign aspect rather than the element-wise adaptive step sizes.

7. Conclusion

We have argued that ADAM combines two components: taking signs and variance adaptation. Our experiments show that the sign aspect is by far the dominant one, but its usefulness is problem-dependent. Our theoretical analysis suggests that it depends on the interplay of stochasticity, the conditioning of the problem, and its axis-alignment. Sign-based methods also seem to have an adverse effect on the generalization performance of the obtained solution; a possible starting point for further research into the generalization effects of optimization algorithms.

The second aspect, variance adaptation, is not restricted to ADAM but can be applied to any update direction. We have provided a general motivation for variance adaptation factors that is independent of the update direction. In particular, we introduced M-SVAG, a variance-adapted variant of momentum SGD, which is a useful addition to the practitioner's toolbox for problems where sign-based methods like ADAM fail. A TensorFlow (Abadi et al., 2015) implementation can be found at `https://github.com/lballes/msvag`.

Acknowledgements

The authors thank Maren Mahsereci for helpful discussions. Lukas Balles kindly acknowledges the support of the International Max Planck Research School for Intelligent Systems (IMPRS-IS).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Amari, S.-I. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

Becker, S. and LeCun, Y. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 Connectionist Models Summer School*, pp. 29–37, 1988.

Chaudhari, P., Choromanska, A., Soatto, S., and LeCun, Y. Entropy-SGD: Biasing gradient descent into wide valleys. *The International Conference on Learning Representations (ICLR)*, 2017.

Defazio, A., Bach, F., and Lacoste-Julien, S. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems 27*, pp. 1646–1654, 2014.

Diaconis, P. and Shahshahani, M. The subgroup algorithm for generating uniform random variables. *Probability in the Engineering and Informational Sciences*, 1(01): 15–32, 1987.

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

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pp. 315–323, 2013.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the Polyak-Lojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811. Springer, 2016.

Kingma, D. and Ba, J. ADAM: A method for stochastic optimization. *The International Conference on Learning Representations (ICLR)*, 2015.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Martens, J. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

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

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

Riedmiller, M. and Braun, H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pp. 586–591. IEEE, 1993.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Schaul, T., Zhang, S., and LeCun, Y. No more pesky learning rates. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 343–351, 2013.

Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

Tieleman, T. and Hinton, G. RMSPROP: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning, Lecture 6.5, 2012.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems 30*, pp. 4151–4161, 2017.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12, September 2016.

Zeiler, M. D. ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.