Escaping Saddles with Stochastic Gradients

Hadi Daneshmand [*1] **Jonas Kohler** [*1] **Aurelien Lucchi** [1] **Thomas Hofmann** [1]

Abstract

We analyze the variance of stochastic gradients along negative curvature directions in certain non-convex machine learning models and show that stochastic gradients exhibit a strong component along these directions. Furthermore, we show that - contrary to the case of isotropic noise - this variance is proportional to the magnitude of the corresponding eigenvalues and not decreasing in the dimensionality. Based upon this observation we propose a new assumption under which we show that the injection of explicit, isotropic noise usually applied to make gradient descent escape saddle points can successfully be replaced by a simple SGD step. Additionally - and under the same condition - we derive the first convergence rate for plain SGD to a *second-order* stationary point in a number of iterations that is independent of the problem dimension.

1. Introduction

In this paper we analyze the use of gradient descent (GD) and its stochastic variant (SGD) to minimize objectives of the form

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \left[f(\mathbf{w}) := \mathbf{E}_{\mathbf{z}\sim\mathcal{P}} \left[f_{\mathbf{z}}(\mathbf{w}) \right] \right], \qquad (1)$$

where $f \in C^2(\mathbb{R}^d, \mathbb{R})$ is a not necessarily convex loss function and $\mathcal{P}$ is an arbitrary probability distribution.

In the era of big data and deep neural networks, (stochastic) gradient descent is a core component of many training algorithms (Bottou, 2010). What makes SGD so attractive is its simplicity, its seemingly universal applicability and a convergence rate that is independent of the size of the training set. One specific trait of SGD is the inherent noise, originating from sampling training points, whose variance has to be controlled in order to guarantee convergence either through a conservative step size (Nesterov, 2013) or via explicit variance-reduction techniques (Johnson & Zhang, 2013).

While the convergence behavior of SGD is well-understood for convex functions (Bottou, 2010), we are here interested in the optimization of non-convex functions which pose additional challenges for optimization in particular due to the presence of saddle points and suboptimal local minima (Dauphin et al., 2014; Choromanska et al., 2015). For example, finding the global minimum of even a degree 4 polynomial can be NP-hard (Hillar & Lim, 2013). Instead of aiming for a global minimizer, a more practical goal is to search for a local optimum of the objective. In this paper we thus focus on reaching a second-order stationary point of smooth non-convex functions. Formally, we aim to find an (ϵ_g, ϵ_h)-second-order stationary point $\mathbf{w}$ such that the following conditions hold:

$$\|\nabla f(\mathbf{w})\| \leq \epsilon_g \quad \text{and} \quad \nabla^2 f(\mathbf{w}) \succcurlyeq -\epsilon_h \mathbf{I}, \qquad (2)$$

where $\epsilon_g, \epsilon_h > 0$.

Existing work, such as (Ge et al., 2015; Jin et al., 2017a), proved convergence to a point satisfying Eq. (2) for modified variants of gradient descent and its stochastic variant by requiring additional noise to be explicitly added to the iterates along the entire path (former) or whenever the gradient is sufficiently small (latter). Formally, this yields the following update step for the perturbed GD and SGD versions:

$$\text{PGD:} \quad \mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla f(\mathbf{w}_t) + r\zeta_{t+1} \qquad (3)$$

$$\text{PSGD:} \quad \mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \left(\nabla f_{\mathbf{z}}(\mathbf{w}_t) + \zeta_t \right), \qquad (4)$$

where ζ_t is typically zero-mean noise sampled uniformly from a unit sphere.

Isotropic noise The perturbed variants of GD and SGD in Eqs. (3)-(4) have been analyzed for the case where the added noise ζ_t is isotropic (Ge et al., 2015; Levy, 2016; Jin et al., 2017a) or at least exhibits a certain amount of variance along all directions in $\mathbb{R}^d$ (Ge et al., 2015). As shown in Table 1, an immediate consequence of such conditions is that they introduce a dependency on the input dimension d in the convergence rate. Furthermore, it is unknown as of today, if this condition is satisfied by the intrinsic noise of vanilla SGD for any specific class of machine learning

[*]Equal contribution [1]ETH, Zurich, Switzerland. Correspondence to: Hadi Daneshmand <hadi.daneshmand@inf.ethz.ch>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

models. Recent empirical observations show that this is not the case for training neural networks (Chaudhari & Soatto, 2017).

In this work, we therefore turn our attention to the following question. Do we need to perturb iterates along *all* dimensions in order for (S)GD to converge to a second-order stationary point? Or is it enough to simply rely on the inherent variance of SGD induced by sampling? More than a purely theoretical exercise, this question has some very important practical implications since in practice the vast majority of existing SGD methods do not add additional noise and therefore do not meet the requirement of isotropic noise. Thus we instead focus our attention on a less restrictive condition for which perturbations only have a guaranteed variance along directions of negative curvature of the objective, i.e. along the eigenvector(s) associated with the minimum eigenvalue of the Hessian. Instead of explicitly adding noise as done in Eqs. (3) and (4), we will from now on consider the simple SGD step:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f_{\mathbf{z}}(\mathbf{w}_t) \tag{5}$$

and propose the following sufficient condition on the stochastic gradient $\nabla f_{\mathbf{z}}(\mathbf{w})$ to guarantee convergence to a second-order stationary point.

Assumption 1 (Correlated Negative Curvature (CNC))**.** *Let $\mathbf{v}_{\mathbf{w}}$ be the eigenvector corresponding to the minimum eigenvalue of the Hessian matrix $\nabla^2 f(\mathbf{w})$. The stochastic gradient $\nabla f_{\mathbf{z}}(\mathbf{w})$ satisfies the CNC assumption, if the second moment of its projection along the direction $\mathbf{v}_{\mathbf{w}}$ is uniformly bounded away from zero, i.e.*

$$\exists \gamma > 0 \text{ s.t. } \forall \mathbf{w} : \mathbf{E}[\langle \mathbf{v}_{\mathbf{w}}, \nabla f_{\mathbf{z}}(\mathbf{w}) \rangle^2] > \gamma \,. \tag{6}$$

Contributions Our contribution is fourfold: First, we analyze the convergence of GD perturbed by SGD steps (Algorithm 1). Under the CNC assumption, we demonstrate that this method converges to an $(\epsilon, \epsilon^{2/5})$-second-order stationary point in $\tilde{\mathcal{O}}(\epsilon^{-2})$ iterations and with high probability. Second, we prove that vanilla SGD as stated in Algorithm 2 -again under Assumption 1- also convergences to an $(\epsilon, \epsilon^{2/5})$-second-order stationary point in $\tilde{\mathcal{O}}(\epsilon^{-4})$ iterations and with high probability. To the best of our knowledge, this is the first second-order convergence result for SGD without adding additional noise. One important consequence of not relying on isotropic noise is that the rate of convergence becomes independent of the input dimension d. This can be a very significant practical advantage when optimizing deep neural networks that contain millions of trainable parameters. Third, we prove that stochastic gradients satisfy Assumption 1 in the setting of learning half-spaces, which is ubiquitous in machine learning. Finally, we provide experimental evidence suggesting the validity of this condition for training neural networks. In particular we show that, while the variance of uniform noise along eigenvectors corresponding to the most negative eigenvalue decreases as $\mathcal{O}(1/d)$, stochastic gradients have a significant component along this direction independent of the *width* and *depth* of the neural net. When looking at the entire eigenspectrum, we find that this variance increases with the magnitude of the associated eigenvalues. Hereby, we contribute to a better understanding of the success of training deep networks with SGD and its extensions.

2. Background & Related work

Reaching a 1st-order stationary point For smooth non-convex functions, a first-order stationary point satisfying $\|\nabla f(\mathbf{x})\| \leq \epsilon$ can be reached by GD and SGD in $\mathcal{O}(\epsilon^{-2})$ and $\mathcal{O}(\epsilon^{-4})$ iterations respectively (Nesterov, 2013). Recently, it has been shown that GD can be accelerated to find such a point in $\mathcal{O}(\epsilon^{-7/4}\log(\epsilon^{-1}))$ (Carmon et al., 2017).

Reaching a 2nd-order stationary point In order to reach second-order stationary points, existing first-order techniques rely on explicitly adding isotropic noise with a known variance (see Eq. (3)). The key motivation for this step is the insight that the area of attraction to a saddle point constitutes an unstable manifold and thus gradient descent methods are unlikely to get stuck, but if they do, adding noise allows them to escape (Lee et al., 2016). Based upon this observations, recent works prove second-order convergence of normalized GD (Levy, 2016) and perturbed GD (Jin et al., 2017a). The later needs at most $\mathcal{O}(\max\{\epsilon_g^{-2}, \epsilon_h^{-4}\}\log^4(d))$ iterations and is thus the first to achieve a poly-log dependency on the dimensionality. The convergence of SGD with additional noise was analyzed in (Ge et al., 2015) but to the best of our knowledge, no prior work demonstrated convergence of SGD *without* explicitly adding noise.

Using curvature information Since negative curvature signals potential descent directions, it seems logical to apply a second-order method to exploit this curvature direction in order to escape saddle points. Yet, the prototypical Newton's method has no global convergence guarantee and is locally attracted by saddle points and even local maxima (Dauphin et al., 2014). Another issue is the computation (and perhaps storage) of the Hessian matrix, which requires $\mathcal{O}(nd^2)$ operations as well as computing the inverse of the Hessian, which requires $\mathcal{O}(d^3)$ computations.

The first problem can be solved by using trust-region methods that guarantee convergence to a second-order stationary point (Conn et al., 2000). Among these methods, the Cubic Regularization technique initially proposed by (Nesterov & Polyak, 2006) has been shown to achieve the optimal worst-case iteration bound $\mathcal{O}(\max\{\epsilon_g^{-3/2}, \epsilon_h^{-3}\})$ (Cartis et al.,

Algorithm	First-order Complexity	Second-order Complexity	d Dependency
Perturbed SGD (Ge et al., 2015)	$\mathcal{O}(d^p\epsilon_g^{-4})$	$\mathcal{O}(d^p\epsilon_h^{-16})$	poly
SGLD (Zhang et al., 2017)	$\mathcal{O}(d^p\epsilon_g^{-2})$	$\mathcal{O}(d^p\epsilon_h^{-4})$	poly
PGD (Jin et al., 2017a)	$\mathcal{O}(\log^4(d/\epsilon_g)\epsilon_g^{-2})$	$\mathcal{O}(\log^4(d/\epsilon_h)\epsilon_h^{-4})$	poly-log
SGD+NEON (Xu & Yang, 2017)	$\tilde{\mathcal{O}}(\epsilon_g^{-4})$	$\tilde{\mathcal{O}}(\epsilon_g^{-8})$	poly-log
CNC-GD (Algorithm 1)	$\mathcal{O}(\epsilon_g^{-2}\log(1/\epsilon_g))$	$\mathcal{O}(\epsilon_h^{-5}\log(1/\epsilon_h))$	free
CNC-SGD (Algorithm 2)	$\mathcal{O}(\epsilon_g^{-4}\log^2(1/\epsilon_g))$	$\mathcal{O}(\epsilon_h^{-10})\log^2(1/\epsilon_h))$	free

Table 1. Dimension dependency and iteration complexity to reach a second-order stationary point as characterized in Eq. (2). The notation $\mathcal{O}(\cdot)$ hides constant factors and $\tilde{\mathcal{O}}(\cdot)$ hides a poly-logarithmic factor.

2012). The second problem can be addressed by replacing the computation of the Hessian by Hessian-vector products that can be computed efficiently in $\mathcal{O}(nd)$ (Pearlmutter, 1994). This is applied e.g. using matrix-free Lanczos iterations (Curtis & Robinson, 2017; Reddi et al., 2017) or online variants such as Oja's algorithm (Allen-Zhu, 2017). Sub-sampling the Hessian can furthermore reduce the dependence on n by using various sampling schemes (Kohler & Lucchi, 2017; Xu et al., 2017). Finally, (Xu & Yang, 2017) and (Allen-Zhu & Li, 2017) showed that noisy gradient updates act as a noisy Power method allowing to find a negative curvature direction using only first-order information. Despite the recent theoretical improvements obtained by such techniques, first-order methods still dominate for training large deep neural networks. Their theoretical properties are however not perfectly well understood in the general case and we here aim to deepen the current understanding.

3. GD Perturbed by Stochastic Gradients

In this section we derive a converge guarantee for a combination of gradient descent and stochastic gradient steps, as presented in Algorithm 1, for the case where the stochastic gradient sequence meets the CNC assumption introduced in Eq. (6). We name this algorithm CNC-PGD since it is a modified version of the PGD method (Jin et al., 2017a), but use the intrinsic noise of SGD instead of requiring noise isotropy. Our theoretical analysis relies on the following smoothness conditions on the objective function f.

Assumption 2 (Smoothness Assumption)**.** *We assume that the function $f \in C^2(\mathbb{R}^d, \mathbb{R})$ has L-Lipschitz gradients and ρ-Lipschitz Hessians and that each function $f_{\mathbf{z}}$ has an ℓ-bounded gradient.*[1] *W.l.o.g. we further assume that ρ, ℓ, and L are greater than one.*

Note that L-smoothness and ρ-Hessian Lipschitzness are standard assumptions for convergence analysis to a second-order stationary point (Ge et al., 2015; Jin et al., 2017a; Nesterov & Polyak, 2006). The boundedness of the stochastic gradient $\nabla f_{\mathbf{z}}(\mathbf{w})$ is often used in stochastic optimization (Moulines & Bach, 2011).

[1] See Appendix A for formal definitions.

Algorithm 1 CNC-PGD

Input: g_{thres}, t_{thres}, T, η and r
$t_{\text{noise}} \leftarrow -t_{\text{thres}} - 1$
for $t = 1, 2, \dots, T$ **do**
  if $\|\nabla f(\mathbf{w}_t)\|^2 \leq g_{\text{thres}}$ and $t - t_{\text{noise}} \geq t_{\text{thres}}$ **then**
    $\tilde{\mathbf{w}}_t \leftarrow \mathbf{w}_t, t_{\text{noise}} \leftarrow t$ *# used in the analysis*
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - r\nabla f_{\mathbf{z}}(\mathbf{w}_t)$ *# $z \overset{i.i.d}{\sim} \mathcal{P}$*
  else
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta\nabla f(\mathbf{w}_t)$
  end if
end for
return $\widehat{\mathbf{w}}$ uniformly from $\{\mathbf{w}_1, \dots, \mathbf{w}_T\}$

Parameter	Value	Dependency on ϵ
η	$1/L$	Independent
r	$c_1(\delta\gamma\epsilon^{4/5})/(\ell^3 L^2)$	$\mathcal{O}(\epsilon^{4/5})$
ω	$\log(\ell L/(\gamma\delta\epsilon))$	$\mathcal{O}(\log(1/\epsilon))$
t_{thres}	$c_2 L(\sqrt{\rho}\epsilon^{2/5})^{-1}\omega$	$\mathcal{O}(\epsilon^{-2/5}\log(1/\epsilon))$
f_{thres}	$c_3\delta\gamma^2\epsilon^{8/5}/(\ell^2 L)^2$	$\mathcal{O}(\epsilon^{8/5})$
g_{thres}	$f_{\text{thres}}/t_{\text{thres}}$	$\mathcal{O}(\epsilon^2/\log(1/\epsilon))$
T	$4(f(\mathbf{w}_0) - f^*)/(\eta\delta g_{\text{thres}})$	$\mathcal{O}(\epsilon^{-2}\log(1/\epsilon))$

Table 2. Parameters of CNC-PGD. Note that the constants f_{thres} and ω are only needed for the analysis and thus not required to run Algorithm 1. The constant $\delta \in (0, 1)$ comes from the probability statement in Theorem 1. Finally the constants c_1, c_2 and c_3 are independent of the parameters γ,δ, ϵ, ℓ, ρ, and L (see Appendix B for more details).

Parameters The analysis presented below relies on a particular choice of parameters. Their values are set based on the desired accuracy ϵ and presented in Table 2.

3.1. PGD Convergence Result

Theorem 1. *Let the stochastic gradients $\nabla f_{\mathbf{z}}(\mathbf{w}_t)$ in CNC-PGD satisfy Assumption 1 and let f, $f_{\mathbf{z}}$ satisfy Assumption 2. Then Algorithm 1 returns an $\left(\epsilon, \sqrt{\rho}\epsilon^{2/5}\right)$-second-order stationary point with probability at least $(1 - \delta)$ after*

$$\mathcal{O}\left((\ell L)^4(\delta\gamma\epsilon)^{-2}\log\left(\frac{\ell L}{\eta\delta\gamma\epsilon^{2/5}}\right)\right)$$

steps, where $\delta < 1$.

Remark CNC-PGD converges polynomially to a second-order stationary point under Assumption 1. By relying on isotropic noise, (Jin et al., 2017a) prove convergence to a $\left(\epsilon, (\rho\epsilon)^{1/2}\right)$-stationary point in $\tilde{\mathcal{O}}\left(1/\epsilon^2\right)$ steps. The result of Theorem 1 matches this rate in terms of first-order optimality but is worse by an $\epsilon^{-0.1}$-factor in terms of the second-order condition. Yet, we do not know whether our rate is the best achievable rate under the CNC condition and whether having isotropic noise is necessary to obtain a faster rate of convergence. As mentioned previously, a major benefit of employing the CNC condition is that it results in a convergence rate that does not depend on the dimension of the parameter space.[2] Furthermore, we believe that the dependency to γ (Eq. (6)) can be significantly improved.

3.2. Proof sketch of Theorem 1

In order to prove Theorem 1, we consider three different scenarios depending on the magnitude of the gradient and the amount of negative curvature. Our proof scheme is mainly inspired by the analysis of perturbed gradient descent (Jin et al., 2017a), where a deterministic sufficient condition is established for escaping from saddle points (see Lemma 11). This condition is shown to hold in the case of isotropic noise. However, the non-isotropic noise coming from stochastic gradients is more difficult to analyze. Our contribution is to show that a less restrictive assumption on the perturbation noise still allows to escape saddle points. Detailed proofs of each lemma are provided in the Appendix.

Large gradient regime When the gradient is large enough, we can invoke existing results on the analysis of gradient descent for non-convex functions (Nesterov, 2013).

Lemma 1. *Consider a gradient descent step $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\nabla f(\mathbf{w}_t)$ on a L-smooth function f. For $\eta \leq 1/L$ this yields the following function decrease:*

$$f(\mathbf{w}_{t+1}) - f(\mathbf{w}_t) \leq -\frac{\eta}{2}\|\nabla f(\mathbf{w}_t)\|^2. \tag{7}$$

Using the above result, we can guarantee the desired decrease whenever the norm of the gradient is large enough. Suppose that $\|\nabla f(\mathbf{w}_t)\|^2 \geq g_{\text{thres}}$, then Lemma 1 immediately yields

$$f(\mathbf{w}_{t+1}) - f(\mathbf{w}_t) \leq -\frac{\eta}{2} g_{\text{thres}}. \tag{8}$$

Small gradient and sharp negative curvature regime Consider the setting where the norm of the gradient is small, i.e. $\|\nabla f(\mathbf{w}_t)\|^2 \leq g_{\text{thres}}$, but the minimum eigenvalue of the Hessian matrix is significantly less than zero, i.e. $\lambda_{\min}(\nabla^2 f(\mathbf{w})) \ll 0$. In such a case, exploiting Assumption 1 (CNC) provides a guaranteed decrease in the function value after t_{thres} iterations, in expectation.

Lemma 2. *Let Assumptions 1 and 2 hold. Consider perturbed gradient steps (Algorithm 1 with parameters as in Table 2) starting from $\tilde{\mathbf{w}}_t$ such that $\|\nabla f(\tilde{\mathbf{w}}_t)\|^2 \leq g_{thres}$. Assume the Hessian matrix $\nabla^2 f(\tilde{\mathbf{w}}_t)$ has a large negative eigenvalue, i.e.*

$$\lambda_{\min}(\nabla^2 f(\tilde{\mathbf{w}}_t)) \leq -\sqrt{\rho}\epsilon^{2/5}. \tag{9}$$

Then, after t_{thres} iterations the function value decreases as

$$\mathbf{E}\left[f(\mathbf{w}_{t+t_{thres}})\right] - f(\tilde{\mathbf{w}}_t) \leq -f_{thres}, \tag{10}$$

where the expectation is over the sequence $\{\mathbf{w}_k\}_{t+1}^{t+t_{thres}}$.

Small gradient with moderate negative curvature regime Suppose that $\|\nabla f(\mathbf{w}_t)\|^2 \leq g_{\text{thres}}$ and that the absolute value of the minimum eigenvalue of the Hessian is close to zero, i.e. we already reached the desired first- and second-order optimality. In this case, we can guarantee that adding noise will only lead to a limited increase in terms of expected function value.

Lemma 3. *Let Assumptions 1 and 2 hold. Consider perturbed gradient steps (Algorithm 1 with parameters as in Table 2) starting from $\tilde{\mathbf{w}}_t$ such that $\|\nabla f(\tilde{\mathbf{w}}_t)\|^2 \leq g_{thres}$. Then after t_{thres} iterations, the function value cannot increase by more than*

$$\mathbf{E}\left[f(\mathbf{w}_{t+t_{thres}})\right] - f(\tilde{\mathbf{w}}_t) \leq \frac{\eta\delta f_{thres}}{4}, \tag{11}$$

where the expectation is over the sequence $\{\mathbf{w}_k\}_{t+1}^{t+t_{thres}}$.

Joint analysis We now combine the results of the three scenarios discussed so far. Towards this end we introduce the set $\mathcal{S}$ as

$$\mathcal{S} := \{\mathbf{w} \in \mathbb{R}^d \mid \|\nabla f(\mathbf{w})\|^2 \geq g_{\text{thres}} \text{ or } \lambda_{\min}\left(\nabla^2 f(\mathbf{w})\right) \leq -\sqrt{\rho}\epsilon^{2/5}\}.$$

Each of the visited parameters $\mathbf{w}_t, t = 1, \ldots, T$ constitutes a random variable. For each of these random variables, we define the event $\mathcal{A}_t := \{\mathbf{w}_t \in \mathcal{S}\}$. When $\mathcal{A}_t$ occurs, the function value decreases in expectation. Since the number of steps required in the analysis of the large gradient regime and the sharp curvature regime are different, we use an amortized analysis similar to (Jin et al., 2017a) where we consider the per-step decrease [3]. Indeed, when the negative curvature is sharp, then Lemma 2 provides a guaranteed decrease in f which - when normalized per step - yields

$$\frac{\mathbf{E}\left[f(\mathbf{w}_{t+t_{\text{thres}}})\right] - f(\tilde{\mathbf{w}}_t)}{t_{\text{thres}}} \leq -\frac{f_{\text{thres}}}{t_{\text{thres}}} = -\eta g_{\text{thres}}. \tag{12}$$

[2] This result is not in conflict with the dimensionality-dependent lower bound established in (Simchowitz et al., 2017) since they make no initialization assumption as we do in Assumption 1 (CNC).

[3] Note that the amortization technique is here used to simplify the presentation but all our results hold without amortization.

The large gradient norm regime of Lemma 1 guarantees a decrease of the same order and hence

$$\mathbf{E}\left[f(\mathbf{w}_{t+1}) - f(\mathbf{w}_t) \mid \mathcal{A}_t\right] \leq -\frac{\eta}{2} g_{\text{thres}} \tag{13}$$

follows from combining the two results. Let us now consider the case when $\mathcal{A}_t^c$ (complement of $\mathcal{A}_t$) occurs. Then the result of Lemma 3 allows us to bound the increase in terms of function value, i.e.

$$\mathbf{E}\left[f(\mathbf{w}_{t+1}) - f(\mathbf{w}_t) \mid \mathcal{A}_t^c\right] \leq \frac{\eta\delta}{4} g_{\text{thres}}. \tag{14}$$

Probabilistic bound The results established so far have shown that *in expectation* the function value decreases until the iterates reach a second-order stationary point, for which Lemma 3 guarantees that the function value does not increase too much subsequently.[4] This result guarantees visiting a second-order stationary point in T steps (see Table 2). Yet, certifying second-order optimality is slightly more intricate as one would need to know which of the parameters $\{\mathbf{w}_1, \ldots, \mathbf{w}_T\}$ meets the required condition. One solution to address this problem is to provide a high probability statement as suggested in (Jin et al., 2017a) (see Lemma 10). We here follow a similar approach except that unlike the result of (Jin et al., 2017a) that relies on exact function values, our results are valid in expectation. Our solution is to establish a high probability bound by returning one of the visited parameters picked uniformly at random. This approach is often used in stochastic non-convex optimization (Ghadimi & Lan, 2013).

The idea is simple: If the number of steps is sufficiently large, then the results of Lemma (1)-(3) guarantee that the number of times we visit a second-order stationary point is high. Let R be a random variable that determines the ratio of $(\epsilon, \sqrt{\rho}\epsilon^{2/5})$-second-order stationary points visited through the optimization path $\{\mathbf{w}_t\}_{t=1,\ldots,T}$. Formally,

$$R := \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\left(\mathcal{A}_t^c\right), \tag{15}$$

where $\mathbb{1}$ is the indicator function. Let $\mathcal{P}_t$ denote the probability of event $\mathcal{A}_t$ and $1 - \mathcal{P}_t$ be the probability of its complement $\mathcal{A}_t^c$. The probability of returning a second-order stationary point is simply

$$\mathbf{E}\left[R\right] = \frac{1}{T}\sum_{t=1}^{T}(1 - \mathcal{P}_t). \tag{16}$$

[4]Since there may exist degenerate saddle points which are second-order stationary but not local minima we cannot guarantee that PGD stays close to a second-order stationary point it visits. One could rule out degenerate saddles using the strict-saddle assumption introduced in (Ge et al., 2015).

Estimating the probabilities $\mathcal{P}_t$ is difficult due to the interdependence of the random variables $\mathbf{w}_t$. However, we can upper bound the sum of the individual $\mathcal{P}_t$'s. Using the law of total expectation and the results from Eq. (13) and (14), we bound the expectation of the function value decrease as:

$$\begin{aligned}\mathbf{E}\left[f(\mathbf{w}_{t+1}) - f(\mathbf{w}_t)\right] \\ \leq \eta g_{\text{thres}}\left(\delta/2 - (1+\delta/2)\mathcal{P}_t\right)/2.\end{aligned} \tag{17}$$

Summing over T iterations yields

$$\begin{aligned}\sum_{i=1}^{T} \mathbf{E}\left[f(\mathbf{w}_{t+1})\right] - \mathbf{E}\left[f(\mathbf{w}_t)\right] \\ \leq \eta g_{\text{thres}}\left(\delta T/2 - (1+\delta/2)\sum_{t=1}^{T}\mathcal{P}_t\right)/2,\end{aligned} \tag{18}$$

which, after rearranging terms, leads to the following upper-bound

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{P}_t \leq \frac{\delta}{2} + \frac{2\left(f(\mathbf{w}_0) - f^*\right)}{T\eta g_{\text{thres}}} \leq \delta. \tag{19}$$

Therefore, the probability that $\mathcal{A}_t^c$ occurs uniformly over $\{1, \ldots, T\}$ is lower bounded as

$$\frac{1}{T}\sum_{t=1}^{T}(1 - \mathcal{P}_t) \geq 1 - \delta, \tag{20}$$

which concludes the proof of Theorem 1.

4. SGD without Perturbation

We now turn our attention to the stochastic variant of gradient descent under the assumption that the stochastic gradients fulfill the CNC condition (Assumption 1). We name this method CNC-SGD and demonstrate that it converges to a second-order stationary point without any additional perturbation. Note that in order to provide the convergence guarantee, we periodically enlarge the step size through the optimization process, as outlined in Algorithm 2. This periodic step size increase amplifies the variance along eigenvectors corresponding to the minimum eigenvalue of the Hessian, allowing SGD to exploit the negative curvature in the subsequent steps (using a smaller step size). Increasing the step size is therefore similar to the perturbation step used in CNC-PGD (Algorithm 1). Although this may not be very common in practice, adaptive stepsizes are not unusual in the literature (see e.g. (Goyal et al., 2017)).

Parameters The analysis of CNC-SGD relies on the particular choice of parameters presented in Table 3.

Algorithm 2 CNC-SGD

Input: t_{thres}, r, η, and T $\quad (\eta < r)$
for $t = 1, 2, \dots, T$ **do**
  if $(t \mod t_{\text{thres}}) = 0$ **then**
    $\tilde{\mathbf{w}}_t \leftarrow \mathbf{w}_t$ *# used in the analysis*
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - r\nabla f_{\mathbf{z}}(\mathbf{w}_t)$ $\quad \# z \overset{i.i.d}{\sim} \mathcal{P}$
  else
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta\nabla f_{\mathbf{z}}(\mathbf{w}_t)$ $\quad \# z \overset{i.i.d}{\sim} \mathcal{P}$
  end if
end for
return $\mathbf{w}_t$ uniformly from $\{\mathbf{w}_1, \dots, \mathbf{w}_T\}$.

Parameter	Value	Dependency to ϵ
r	$c_1\delta\gamma\epsilon^{4/5}/(\ell^3 L)$	$\mathcal{O}(\epsilon^{4/5})$
f_{thres}	$c_2\delta\gamma^2\epsilon^{8/5}/(\ell^4 L)$	$\mathcal{O}(\epsilon^{8/5})$
ω	$c_3\log(\ell L/(\eta\epsilon r))$	$\mathcal{O}(\log(1/\epsilon))$
η	$c_4\delta^2\gamma^2\epsilon^2/(\ell^6 L^2\omega)$	$\mathcal{O}(\epsilon^2/\log(1/\epsilon))$
t_{thres}	$(\eta\epsilon^{2/5})^{-1}\omega$	$\mathcal{O}(\epsilon^{-12/5}\log^2(1/\epsilon))$
g_{thres}	$f_{\text{thres}}/t_{\text{thres}}$	$\mathcal{O}(\epsilon^4/\log^2(1/\epsilon))$
T	$4(f(\mathbf{w}_0) - f^*)/(\delta g_{\text{thres}})$	$\mathcal{O}(\epsilon^{-4}\log^2(1/\epsilon))$

Table 3. Parameters of CNC-SGD: the parameters f_{thres} and g_{thres} are used exclusively in the analysis and are thus not needed to run the algorithm. The constants $c_1, c_2, \dots, c_4$ are independent of the parameters γ,δ, ϵ, ρ, and L (see Appendix B for more details).

Theorem 2. *Let the stochastic gradients $\nabla f_{\mathbf{z}}(\mathbf{w}_t)$ in CNC-SGD satisfy Assumption 1 and let f, $f_{\mathbf{z}}$ satisfy Assumption 2. Then Algorithm 2 returns an $\left(\epsilon, \sqrt{\rho}\epsilon^{2/5}\right)$-second-order stationary point with probability at least $(1-\delta)$ after*

$$\mathcal{O}\left(\left(\frac{\delta\gamma\epsilon}{L\ell^{5/2}}\right)^{-4}\log^2\left(\frac{\ell L}{\epsilon\delta\gamma}\right)\right)$$

steps, where $\delta < 1$.

Remarks As reported in Table 1, perturbed SGD - with isotropic noise - converges to an $(\epsilon, \epsilon^{1/4})$-second-order stationary point in $\mathcal{O}(d^p\epsilon^{-4})$ steps (Ge et al., 2015). Here, we prove that under the CNC assumption, vanilla SGD - i.e. without perturbations - converges to an $(\epsilon, \sqrt{\rho}\epsilon^{2/5})$second-order stationary point using $\tilde{\mathcal{O}}(\epsilon^{-4})$ stochastic gradient steps. Our result matches the result of (Ge et al., 2015) in terms of first-order optimality and yields an improvement by an $\epsilon^{0.15}$-factor in terms of second-order optimality. However, this second-order optimality rate is still worse by an $\epsilon^{-0.1}$-factor compared to the best known convergence rate for perturbed SGD established by (Zhang et al., 2017), which requires $\mathcal{O}(d^p\epsilon^{-4})$ iterations for an $(\epsilon, \epsilon^{1/2})$-second-order stationary point. One can even improve the convergence guarantee of SGD by using the NEON framework (Allen-Zhu & Li, 2017; Xu & Yang, 2017) but a perturbation with isotropic noise is still required. The theoretical guarantees we provide in Theorem 2, however, are based on a less restrictive assumption. As we prove in the following Section, this assumption actually holds for stochastic gradients when learning half-spaces. Subsequently, in Section 6, we present empirical observations that suggest its validity even for training wide and deep neural networks.

5. Learning Half-spaces with Correlated Negative Curvature

The analysis presented in the previous sections relies on the CNC assumption introduced in Eq. (6). As mentioned before, this assumption is weaker than the isotropic noise condition required in previous work. In this Section we confirm the validity of this condition for the problem of learning half-spaces which is a core problem in machine learning, commonly encountered when training Perceptrons, Support Vector Machines or Neural Networks (Zhang et al., 2015). Learning a half-space reduces to a minimization problem of the following form

$$\min_{\mathbf{w}\in\mathbb{R}^d}\left[f(\mathbf{w}) := \mathbf{E}_{\mathbf{z}\sim\mathcal{P}}\left[\varphi(\mathbf{w}^\top\mathbf{z})\right]\right], \tag{21}$$

where φ is an arbitrary loss function and the data distribution $\mathcal{P}$ might have a finite or infinite support. There are different choices for the loss function φ, e.g. zero-one loss, sigmoid loss or piece-wise linear loss (Zhang et al., 2015). Here, we assume that $\varphi(\cdot)$ is differentiable. Generally, the objective $f(\mathbf{w})$ is non-convex and might exhibit many local minima and saddle points. Note that the stochastic gradient is unbiased and defined as

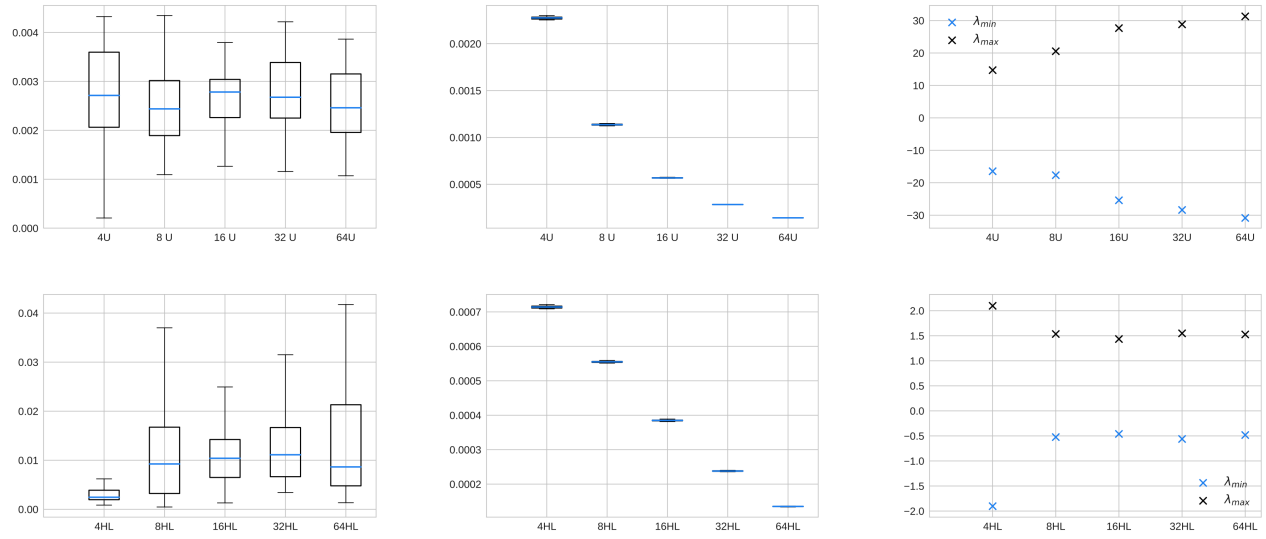
$$\nabla f_{\mathbf{z}}(\mathbf{w}) = \varphi'(\mathbf{w}^\top\mathbf{z})\mathbf{z}, \quad \nabla f(\mathbf{w}) = \mathbf{E}_{\mathbf{z}}\left[\nabla f_{\mathbf{z}}(\mathbf{w})\right], \tag{22}$$

where the samples $\mathbf{z}$ are drawn from the distribution $\mathcal{P}$.

Noise isotropy vs. CNC assumption. First, one can easily find a scenario where the noise isotropy condition is violated for stochastic gradients. Take for example the case where the data distribution from which $\mathbf{z}$ is sampled lives in a low-dimensional space $\mathcal{L} \subset \mathbb{R}^d$. In this case, one can prove that there exists a vector $\mathbf{u} \in \mathbb{R}^d$ orthogonal to all $\mathbf{z} \in \mathcal{L}$. Then clearly $\mathbf{E}\left[(\mathbf{u}^\top\nabla f_{\mathbf{z}}(\mathbf{w}))^2\right] = 0$ and thus $\nabla f_{\mathbf{z}}(\mathbf{w})$ does not have components along all directions.

However - under mild assumptions - we show that the stochastic gradients do have a significant component along directions of negative curvature. Lemma 4 makes this argument precise by establishing a lower bound on the second moment of the stochastic gradients projected onto eigenvectors corresponding to negative eigenvalues of the Hessian matrix $\nabla^2 f(\mathbf{w})$. To establish this lower bound we require the following structural property of the loss function φ.

Assumption 3. *Suppose that the magnitude of the second-order derivative of φ is bounded by a constant factor of its*



1. Stochastic gradients 2. Isotropic noise 3. Extreme eigenvalues

Figure 1. Average variance of stochastic gradients (1) and isotropic noise (2) along eigenvectors corresponding to λ_{min} and extreme eigenvalues (3) of 30 random weight settings in a 1-Layer Neural Network with increasing number of units U (top) and multi-layer Neural Network with increasing number of hidden layers HL (bottom).

first-order derivative, i.e.

$$|\varphi''(\alpha)| \leq c|\varphi'(\alpha)| \tag{23}$$

holds for all α in the domain of φ and $c > 0$.

The reader might notice that this condition resembles the self-concordant assumption often used in the optimization literature (Nesterov, 2013), for which the second derivative is bounded by the third derivative. One can easily check that this condition is fulfilled by commonly used activation functions in neural networks, such as the sigmoid and softplus. We now leverage this property to prove that the stochastic gradient $\nabla f_{\mathbf{z}}(\mathbf{w})$ satisfies Assumption 1 (CNC).

Lemma 4. *Consider the problem of learning half-spaces as stated in Eq. (21), where φ satisfies Assumption 3. Furthermore, assume that the support of $\mathcal{P}$ is a subset of the unit sphere.*[5] *Let $\mathbf{v}$ be a unit length eigenvector of $\nabla^2 f(\mathbf{w})$ with corresponding eigenvalue $\lambda < 0$. Then*

$$\mathbf{E}_{\mathbf{z}}\left[(\nabla f_{\mathbf{z}}(\mathbf{w})^\top \mathbf{v})^2\right] \geq (\lambda/c)^2. \tag{24}$$

Discussion Since the result of Lemma 4 holds for any eigenvector $\mathbf{v}$ associated with a negative eigenvalue $\lambda < 0$, this naturally includes the eigenvector(s) corresponding to λ_{min}. As a result, Assumption 1 (CNC) holds for stochastic gradients on learning half-spaces. Combining this result with the derived convergence guarantees in Theorem 1 implies that a mix of SGD and GD steps (Algorithm 1) obtains a second-order stationary point in polynomial time. Furthermore, according to Theorem 2, vanilla SGD obtains a second-order stationary point in polynomial time without *any* explicit perturbation. Notably, both established convergence guarantees are dimension free.

Furthermore, Lemma 4 reveals an interesting relationship between stochastic gradients and eigenvectors at a certain iterate $\mathbf{w}$. Namely, the variance of stochastic gradients along these vectors scales proportional to the magnitude of the negative eigenvalues within the spectrum of the Hessian matrix. This is in clear contrast to the case of isotropic noise variance which is *uniformly* distributed along all eigenvectors of the Hessian matrix. The difference can be important form a generalization point of view. Consider the simplified setting where φ is square loss. Then the eigenvectors with large eigenvalues correspond to the principal directions of the data. In this regard, having a lower variance along the non-principal directions avoids over-fitting.

In the following section we confirm the above results and furthermore show experiments on Neural Networks that suggest the validity of these results beyond the setting of learning half-spaces.

[5] This assumption is equivalent to assuming the random variable $\mathbf{z}$ lies inside the unit sphere, which is common in learning half-space (Zhang et al., 2015).

6. Experiments

In this Section we first show that vanilla SGD (Algorithm 2) as well as GD with a stochastic gradient step as perturbation (Algorithm 1) indeed escape saddle points. Towards this end, we initialize SGD, GD, perturbed GD with isotropic noise (ISO-PGD) (Jin et al., 2017a) and CNC-PGD close to a saddle point on a low dimensional learning-halfspaces problem with Gaussian input data and sigmoid loss. Figure 2 shows suboptimality over epochs for an average of 10 runs. The results are in line with our analysis since all stochastic methods quickly find a negative curvature direction to escape the saddle point. See Appendix E for more details.[6]

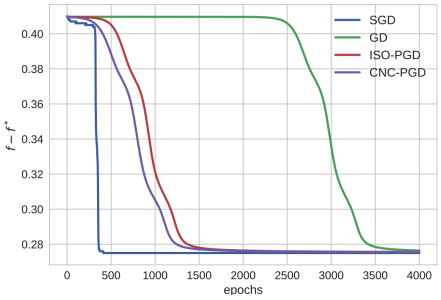


Figure 2. Learning halfspaces ($n = 40, d = 4$): The stochastic methods need less iterations to escape the saddle.

Secondly - and more importantly - we study the properties of the variance of stochastic gradients depending on the width and depth of neural networks. All of these experiments are conducted using feed-forward networks on the well-known MNIST classification task ($n = 70'000$). Specifically, we draw $m = 30$ random parameters $\mathbf{w}_i$ in each of these networks and test Assumption 1 by estimating the second moment of the stochastic gradients projected onto the eigenvectors $\mathbf{v}_k$ of $\nabla^2 f(\mathbf{w}_i)$ as follows

$$\mu_k = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{n}\sum_{j=1}^{n}\left(\nabla f_j(\mathbf{w}_i)^\top \mathbf{v}_k\right)^2\right). \tag{25}$$

We do the same for n isotropic noise vectors drawn from the unit ball $\mathcal{B}^d$ around each $\mathbf{w}_i$.[7] Figure 1 shows this estimate for eigenvectors corresponding to the minimum eigenvalues for a 1 hidden layer network with increasing number of units (top) and for a 10 hidden unit network with increasing number of layers (bottom). Similar results on the entire negative eigenspectrum can be found in Appendix E. Figure 3 shows how μ_k varies with the magnitude of the corresponding negative eigenvalues λ_k. Again we evaluate 30 random parameter settings in neural networks with

[6]Rather than an encompassing benchmark of the different methods, this result is to be seen as a proof of concept.

[7]For a fair comparison all involved vectors were normalized.

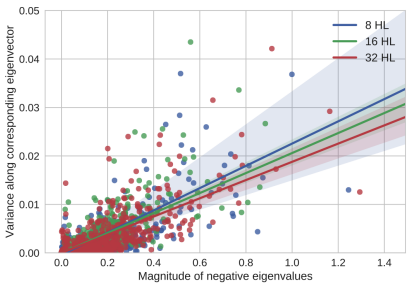


Figure 3. Variance of stochastic gradients along eigenvectors corresponding to eigenvalues of different magnitudes computed on neural networks with 8, 16 and 32 hidden layers. Scatterplot and fitted linear model with 95% confidence interval.

increasing depth. Two conclusions can be drawn from the results: (i) Although the variance of isotropic noise along eigenvectors corresponding to $\lambda_{\min}$ decreases as $\mathcal{O}(1/d)$, the stochastic gradients maintain a significant component along the directions of most negative curvature independent of *width* and *depth* of the neural network (see Figure 1), (ii) the stochastic gradients yield an increasing variance along eigenvectors corresponding to larger eigenvalues (see Figure 3). These findings suggest important implications. (i) justify the use and explain the success of training wide and deep neural networks with pure SGD despite the presence of saddle points. (ii) suggests that the bound established in Lemma 4 may well be extended to more general settings such as training neural networks and illustrates the implicit regularization of optimization methods that rely on stochastic gradients since directions of large curvature correspond to principal (more robust) components of the data for many machine learning models.

7. Conclusion

In this work we have analyzed the convergence of PGD and SGD for optimizing non-convex functions under a new assumption -named CNC - that requires the stochastic noise to exhibit a certain amount of variance along the directions of most negative curvature. This is a less restrictive assumption than the noise isotropy condition required by previous work which causes a dependency to the problem dimensionality in the convergence rate. We have shown theoretically that stochastic gradients satisfy the CNC assumption and reveal a variance proportional to the eigenvalue's magnitude for the problem of learning half-spaces. Furthermore, we provided empirical evidence that suggests the validity of this assumption in the context of neural networks and thus contributes to a better understanding of training these models with stochastic gradients. Proving this observation theoretically and investigating its implications on the optimization and generalization properties of stochastic gradients methods is an interesting direction of future research.

Acknowledgements

We would like to thank Kfir Levy, Gary Becigneul, Yannic Kilcher and Kevin Roth for their helpful discussions. We also thank Antonio Orvieto for pointing out a mistake in an early draft.

References

Allen-Zhu, Z. Natasha 2: Faster non-convex optimization than sgd. *arXiv preprint arXiv:1708.08694*, 2017.

Allen-Zhu, Z. and Li, Y. Neon2: Finding local minima via first-order oracles. *arXiv preprint arXiv:1711.06673*, 2017.

Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Carmon, Y., Hinder, O., Duchi, J. C., and Sidford, A. " convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. *arXiv preprint arXiv:1705.02766*, 2017.

Cartis, C., Gould, N. I., and Toint, P. L. *How Much Patience to You Have?: A Worst-case Perspective on Smooth Noncovex Optimization*. Science and Technology Facilities Council Swindon, 2012.

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

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

Conn, A. R., Gould, N. I., and Toint, P. L. *Trust region methods*. SIAM, 2000.

Curtis, F. E. and Robinson, D. P. Exploiting negative curvature in deterministic and stochastic optimization. *arXiv preprint arXiv:1703.00412*, 2017.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pp. 2933–2941, 2014.

Ge, R., Huang, F., Jin, C., and Yuan, Y. Escaping from saddle points-online stochastic gradient for tensor decomposition. In *COLT*, pp. 797–842, 2015.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Hillar, C. J. and Lim, L.-H. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. *arXiv preprint arXiv:1703.00887*, 2017a.

Jin, C., Netrapalli, P., and Jordan, M. I. Accelerated gradient descent escapes saddle points faster than gradient descent. *arXiv preprint arXiv:1711.10456*, 2017b.

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

Kohler, J. M. and Lucchi, A. Sub-sampled cubic regularization for non-convex optimization. In *International Conference on Machine Learning*, 2017.

Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient descent converges to minimizers. *arXiv preprint arXiv:1602.04915*, 2016.

Levy, K. Y. The power of normalization: Faster evasion of saddle points. *arXiv preprint arXiv:1611.04831*, 2016.

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

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

Nesterov, Y. and Polyak, B. T. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Pearlmutter, B. A. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

Reddi, S. J., Zaheer, M., Sra, S., Poczos, B., Bach, F., Salakhutdinov, R., and Smola, A. J. A generic approach for escaping saddle points. *arXiv preprint arXiv:1709.01434*, 2017.

Simchowitz, M., Alaoui, A. E., and Recht, B. On the gap between strict-saddles and true convexity: An omega (log d) lower bound for eigenvector approximation. *arXiv preprint arXiv:1704.04548*, 2017.

Xu, P., Roosta-Khorasani, F., and Mahoney, M. W. Newton-type methods for non-convex optimization under inexact hessian information. *arXiv preprint arXiv:1708.07164*, 2017.

Xu, Y. and Yang, T. First-order stochastic algorithms for escaping from saddle points in almost linear time. *arXiv preprint arXiv:1711.01944*, 2017.

Zhang, Y., Lee, J. D., Wainwright, M. J., and Jordan, M. I. Learning halfspaces and neural networks with random initialization. *arXiv preprint arXiv:1511.07948*, 2015.

Zhang, Y., Liang, P., and Charikar, M. A hitting time analysis of stochastic gradient langevin dynamics. In *COLT*, 2017.