A Quantitative Analysis of the Effect of Batch Normalization on Gradient Descent

Yongqiang Cai[1] Qianxiao Li[1 2] Zuowei Shen[1]

Abstract

Despite its empirical success and recent theoretical progress, there generally lacks a quantitative analysis of the effect of batch normalization (BN) on the convergence and stability of gradient descent. In this paper, we provide such an analysis on the simple problem of ordinary least squares (OLS), where the precise dynamical properties of gradient descent (GD) is completely known, thus allowing us to isolate and compare the additional effects of BN. More precisely, we show that unlike GD, gradient descent with BN (BNGD) converges for arbitrary learning rates for the weights, and the convergence remains linear under mild conditions. Moreover, we quantify two different sources of acceleration of BNGD over GD – one due to over-parameterization which improves the effective condition number and another due having a large range of learning rates giving rise to fast descent. These phenomena set BNGD apart from GD and could account for much of its robustness properties. These findings are confirmed quantitatively by numerical experiments, which further show that many of the uncovered properties of BNGD in OLS are also observed qualitatively in more complex supervised learning problems.

1. Introduction

Batch normalization (BN) is one of the most important techniques for training deep neural networks and has proven extremely effective in avoiding gradient blowups during back-propagation and speeding up convergence. In its original introduction (Ioffe & Szegedy, 2015), the desirable effects of BN are attributed to the so-called "reduction of covariate shift". However, it is unclear what this statement means in precise mathematical terms.

Although recent theoretical work have established certain convergence properties of gradient descent with BN (BNGD) and its variants (Ma & Klabjan, 2017; Kohler et al., 2018; Arora et al., 2019), there generally lacks a quantitative comparison between the dynamics of the usual gradient descent (GD) and BNGD. In other words, a basic question that one could pose is: what quantitative changes does BN bring to the stability and convergence of gradient descent dynamics? Or even more simply: why should one use BNGD instead of GD? To date, a general mathematical answer to these questions remain elusive. This can be partly attributed to the complexity of the optimization objectives that one typically applies BN to, such as those encountered in deep learning. In these cases, even a quantitative analysis of the dynamics of GD itself is difficult, not to mention a precise comparison between the two.

For this reason, it is desirable to formulate the simplest nontrivial setting, on which one can concretely study the effect of batch normalization and answer the questions above in a quantitative manner. This is the goal of the current paper, where we focus on perhaps the simplest supervised learning problem – ordinary least squares (OLS) regression – and analyze precisely the effect of BNGD when applied to this problem. A primary reason for this choice is that the dynamics of GD in least-squares regression is completely understood, thus allowing us to isolate and contrast the additional effects of batch normalization.

Our main findings can be summarized as follows

1. Unlike GD, BNGD converges for arbitrarily large learning rates for the weights, and the convergence remains linear under mild conditions.
2. The asymptotic linear convergence of BNGD is faster than that of GD, and this can be attributed to the overparameterization that BNGD introduces.
3. Unlike GD, the convergence rate of BNGD is insensitive to the choice of learning rates. The range of insensitivity can be characterized, and in particular it

[1] Department of Mathematics, National University of Singapore, Singapore [2] Institute of High Performance Computing, A*STAR, Singapore. Correspondence to: Yongqiang Cai <matcyon@nus.edu.sg>, Qianxiao Li <Qianxiao@nus.edu.sg>, Zuowei Shen <matzuows@nus.edu.sg>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

increases with the dimensionality of the problem.

Although these findings are established concretely only for the OLS problem, we will show through numerical experiments that some of them hold qualitatively, and sometimes even quantitatively for more general situations in deep learning.

1.1. Related Work

Batch normalization was originally introduced in Ioffe & Szegedy (2015) and subsequently studied in further detail in Ioffe (2017). Since its introduction, it has become an important practical tool to improve stability and efficiency of training deep neural networks (Bottou et al., 2018). Initial heuristic arguments attribute the desirable features of BN to concepts such as "covariate shift", but alternative explanations based on landscapes (Santurkar et al., 2018) and effective regularization (Bjorck et al., 2018) have been proposed.

Recent theoretical studies of BN include Ma & Klabjan (2017); Kohler et al. (2018); Arora et al. (2019). We now outline the main differences between them and the current work. In Ma & Klabjan (2017), the authors proposed a variant of BN, the diminishing batch normalization (DBN) algorithm and established its convergence to a stationary point of the loss function. In Kohler et al. (2018), the authors also considered a BNGD variant by dynamically setting the learning rates and using bisection to optimize the rescaling variables introduced by BN. It is shown that this variant of BNGD converges linearly for simplified models, including an OLS model and "learning halfspaces". The primary difference in the current work is that we do not dynamically modify the learning rates, and consider instead a constant learning rate, i.e. the original BNGD algorithm. This is an important distinction; While a decaying or dynamic learning rate is sometimes used in GD, in the case of BN it is critical to analyze the constant learning rate case, precisely because one of the key practical advantages of BN is that a big learning rate can be used. Moreover, this allows us to isolate the influence of batch normalization itself, without the potentially obfuscating effects a dynamic learning rate schedule can introduce (e.g. see Eq. (10) and the discussion that follows). As the goal of considering a simplified model is to analyze the additional effects purely due to BN on GD, it is desirable to perform our analysis in this regime.

In Arora et al. (2019), the authors proved a general convergence result for BNGD of $\mathcal{O}(k^{-1/2})$ in terms of the gradient norm for objectives with Lipschitz continuous gradients. This matches the best result for gradient descent on general non-convex functions with learning rate tuning (Carmon et al., 2017). In contrast, our convergence result is in iteration and is shown to be linear under mild conditions (Theorem 3.4). This convergence result is stronger, but this is to be expected since we are considering a specific case. More importantly, we discuss concretely how BNGD offers advantages over GD instead of just matching its best-case performance. For example, not only do we show that convergence occurs for any learning rate, we also derive a quantitative relationship between the learning rate and the convergence rate, from which the robustness of BNGD on OLS can be explained (see Section 3).

1.2. Organization

Our paper is organized as follows. In Section 2, we outline the ordinary least squares (OLS) problem and present GD and BNGD as alternative means to solve this problem. In Section 3, we demonstrate and analyze the convergence of the BNGD for the OLS model, and in particular contrast the results with the behavior of GD, which is completely known for this model. We also discuss the important insights to BNGD that these results provide us with. We then validate these findings on more general supervised learning problems in Section 4. Finally, we conclude in Section 5.

2. Background

2.1. Ordinary Least Squares and Gradient Descent

Consider the simple linear regression model where $x \in \mathbb{R}^d$ is a random input column vector and y is the corresponding output variable. Since batch normalization is applied for each feature separately, in order to gain key insights it is sufficient to consider the case $y \in \mathbb{R}$. A noisy linear relationship is assumed between the dependent variable y and the independent variables x, i.e. $y = x^T w + \text{noise}$ where $w \in \mathbb{R}^d$ is the vector of trainable parameters. Denote the following moments:

$$H := E[xx^T], \quad g := E[xy], \quad c := E[y^2]. \tag{1}$$

To simplify the analysis, we assume the covariance matrix H of x is positive definite and the mean $E[x]$ of x is zero. The eigenvalues of H are denoted as $\lambda_i(H), i = 1, 2, ...d,$. Particularly, the maximum and minimum eigenvalue of H is denoted by λ_{max} and λ_{min} respectively. The condition number of H is defined as $\kappa := \frac{\lambda_{max}}{\lambda_{min}}$. Note that the positive definiteness of H allows us to define the vector norm $\|.\|_H$ by $\|x\|_H^2 = x^T Hx$.

The ordinary least squares (OLS) method for estimating the unknown parameters w leads to the following optimization problem,

$$\begin{aligned} \min_{w\in\mathbb{R}^d} J_0(w) :&= \tfrac{1}{2} E_{x,y}[(y - x^T w)^2] \\ &= \tfrac{c}{2} - w^T g + \tfrac{1}{2} w^T Hw, \end{aligned} \tag{2}$$

which has unique minimizer $w = u := H^{-1}g$.

The gradient descent (GD) method (with step size or learning rate ε) for solving the optimization problem (2) is given by the iteration

$$w_{k+1} = w_k - \varepsilon \nabla_w J_0(w_k) = (I - \varepsilon H) w_k + \varepsilon g, \quad (3)$$

which converges if $0 < \varepsilon < \frac{2}{\lambda_{max}} =: \varepsilon_{max}$, and the convergence rate is determined by the spectral radius $\rho_\varepsilon := \rho(I - \varepsilon H) = \max_i\{|1 - \varepsilon \lambda_i(H)|\}$ with

$$\|u - w_{k+1}\| \le \rho(I - \varepsilon H)\|u - w_k\|. \quad (4)$$

It is well-known (e.g. see Chapter 4 of Saad (2003)) that the optimal learning rate is $\varepsilon_{opt} = \frac{2}{\lambda_{max}+\lambda_{min}}$, where the optimal convergence rate is $\rho_{opt} = \frac{\kappa - 1}{\kappa + 1}$.

2.2. Batch Normalization

Batch normalization is a feature-wise normalization procedure typically applied to the output, which in this case is simply $z = x^T w$. The normalization transform is defined as follows:

$$N(z) := \frac{z - E[z]}{\sqrt{\text{Var}[z]}} = \frac{x^T w}{\sigma}, \quad (5)$$

where $\sigma := \sqrt{w^T H w}$. After this rescaling, $N(z)$ will be order 1, and hence in order to reintroduce the scale (Ioffe & Szegedy, 2015), we multiply $N(z)$ with a rescaling parameter a (Note that the shift parameter can be set zero since $\mathbb{E}[w^T x|w] = 0$). Hence, we get the BN version of the OLS problem (2):

$$\begin{aligned}\min_{w\in\mathbb{R}^d, a\in\mathbb{R}} J(a, w) :&= \tfrac{1}{2} E_{x,y}\left[\left(y - aN(x^T w)\right)^2\right] \\ &= \tfrac{c}{2} - \tfrac{w^T g}{\sigma} a + \tfrac{1}{2} a^2. \quad (6)\end{aligned}$$

The objective function $J(a, w)$ is no longer convex. In fact, it has critical points, $\{(a^*, w^*)|a^* = 0, w^{*T} g = 0\}$, which are saddle points of $J(a, w)$ if $g \ne 0$.

We are interested in the critical points which constitute the set of global minima and satisfy the relations

$$a^* = \text{sign}(s)\sqrt{u^T H u}, w^* = su, \text{ for some } s \in \mathbb{R} \setminus \{0\}.$$

It is easy to check that they are in fact global minimizers and the Hessian matrix at each point is degenerate. Nevertheless, the saddle points are strict (see appendix B.1), which typically simplifies the analysis of gradient descent on non-convex objectives (Lee et al., 2016; Panageas & Piliouras, 2017).

We consider the gradient descent method for solving the problem (6), which we hereafter call batch normalization gradient descent (BNGD). We set the learning rates for a and w to be ε_a and ε respectively. These may be different, for reasons which will become clear in the subsequent analysis. We thus have the following discrete-time dynamical system:

$$a_{k+1} = a_k + \varepsilon_a\left(\tfrac{w_k^T g}{\sigma_k} - a_k\right), \quad (7)$$

$$w_{k+1} = w_k + \varepsilon \tfrac{a_k}{\sigma_k}\left(g - \tfrac{w_k^T g}{\sigma_k^2} H w_k\right). \quad (8)$$

To simplify subsequent notation, we denote by H^* the matrix

$$H^* := H - \frac{H u u^T H}{u^T H u}, \quad (9)$$

We will see later that the over-parameterization introduced by BN gives rise to a degenerate Hessian matrix $\text{diag}\left(1, \frac{\|u\|^2}{\|w^*\|^2} H^*\right)$ at a minimizer (a^*, w^*), and the BNGD dynamics is governed by H^* instead of H as in the GD case. The matrix H^* is positive semi-definite ($H^* u = 0$) and has better spectral properties than H, such as a lower effective condition number $\kappa^* = \frac{\lambda^*_{max}}{\lambda^*_{min}} \le \kappa$, where λ^*_{max} and λ^*_{min} are the maximal and minimal nonzero eigenvalues of H^* respectively. Particularly, $\kappa^* < \kappa$ for almost all u (see appendix B.1).

3. Mathematical Analysis of BNGD on OLS

In this section, we discuss several mathematical results one can derive concretely for BNGD on the OLS problem (6).

Compared with GD, the update coefficient before $H w_k$ in Eq. (8) changed from ε in Eq. (3) to a complicated term which we call the *effective learning rate* $\hat{\varepsilon}_k$

$$\hat{\varepsilon}_k := \varepsilon \frac{a_k}{\sigma_k} \frac{w_k^T g}{\sigma_k^2}. \quad (10)$$

Also, notice that with the over-parameterization introduced by a, it is no longer necessary for w_k to converge to u. In fact, any non-zero scalar multiple of u can be a global minimum. Hence, instead of considering the residual $u - w_k$ as in the GD analysis Eq. (4), we may combine Eq. (7) and Eq. (8) to give

$$u - \frac{w_k^T g}{\sigma_k^2} w_{k+1} = (I - \hat{\varepsilon}_k H)\left(u - \frac{w_k^T g}{\sigma_k^2} w_k\right). \quad (11)$$

Define the modified residual $e_k := u - (w_k^T g/\sigma_k^2) w_k$, which equals 0 if and only if w_k is a global minimizer. Observe that the mapping $u \mapsto (w^T g/\sigma^2) w = (w^T H u / w^T H w) w$ is an orthogonal projection under the inner product induced by H, hence we immediately have

$$\|e_{k+1}\|_H \le \left\|u - \frac{w_k^T g}{\sigma_k^2} w_{k+1}\right\|_H \le \rho(I - \hat{\varepsilon}_k H)\|e_k\|_H, \quad (12)$$

where $\rho(I - \hat{\varepsilon}_k H)$ is spectral radius of the matrix $I - \hat{\varepsilon}_k H$. In other words, as long as $\max_i\{|1 - \hat{\varepsilon}_k \lambda_i(H)|\} \le \hat{\rho} < 1$

for some $\hat{\rho} < 1$ and all k, we have linear convergence of the residual (which also implies linear convergence of the objective, see appendix Lemma B.22).

At this point, we make an important observation: if we allow for dynamic learning rates, we may simply set $\hat{\varepsilon}_k = c$ for some fixed $c \in (0, 2/\lambda_{max})$ at every iteration. Then, linear convergence is immediate. However, it is clear that this fast convergence is almost entirely due to the effect of dynamic learning rates, and this has limited relevance in explaining the effect of BN. Moreover, comparing with Eq. (4) one can observe that with this choice, BNGD and GD have the same optimal convergence rates, and so this cannot offer explanations for any advantage of BNGD over GD either. For these reasons, it is important to avoid such dynamic learning rate assumptions.

As discussed above, without using dynamic learning rates one has to then estimate $\hat{\varepsilon}_k$ to establish convergence. Heuristically, observe that if ε small enough, this is likely true as the other terms can be controlled due to the normalization. Thus, convergence for small ε should hold. In order to handle the large ε case, we establish a simple but useful scaling law that draws connections amongst cases with different ε scales.

3.1. Scaling Property

The dynamical properties of the BNGD iterations are governed by a set of parameters, or a *configuration* $\{H, u, a_0, w_0, \varepsilon_a, \varepsilon\}$.

Definition 3.1 (Equivalent configuration). *Two configurations, $\{H, u, a_0, w_0, \varepsilon_a, \varepsilon\}$ and $\{H', u', a_0', w_0', \varepsilon_a', \varepsilon'\}$, are said to be equivalent if for BNGD iterates $\{w_k\}$, $\{w_k'\}$ following these configurations respectively, there is an invertible linear transformation T and a nonzero constant t such that $w_k' = Tw_k, a_k' = ta_k$ for all k.*

The scaling property ensures that equivalent configurations must converge or diverge together, with the same rate up to a constant multiple. Now, it is easy to check the system has the following scaling law.

Proposition 3.2 (Scaling property). *Suppose $\mu \neq 0, \gamma \neq 0, r \neq 0, Q^TQ = I$, then (1) The configurations $\{\mu Q^T HQ, \frac{\gamma}{\sqrt{\mu}}Qu, \gamma a_0, \gamma Q w_0, \varepsilon_a, \varepsilon\}$ and $\{H, u, a_0, w_0, \varepsilon_a, \varepsilon\}$ are equivalent. (2) The configurations $\{H, u, a_0, w_0, \varepsilon_a, \varepsilon\}$ and $\{H, u, a_0, rw_0, \varepsilon_a, r^2\varepsilon\}$ are equivalent.*

It is worth noting that the scaling property (2) in Proposition 3.2 originates from the batch-normalization procedure and is independent of the specific structure of the loss function. Hence, it is valid for general problems where BN is used (appendix Lemma A.3). Despite being a simple result, the scaling property is important in determining the dynamics of BNGD, and is useful in our subsequent analysis of its convergence and stability properties. For example, it indicates that separating learning rate for weights (w) and rescaling parameters (a) is equivalent to changing the norm of initial weights.

3.2. Batch Normalization Converges for Arbitrary Step Size

Having established the scaling law, we then have the following convergence result for BNGD on OLS.

Theorem 3.3 (Convergence of BNGD). *The iteration sequence (a_k, w_k) in Eq. (7)-(8) converges to a stationary point for any initial value (a_0, w_0) and any $\varepsilon > 0$, as long as $\varepsilon_a \in (0, 1]$. Particularly, we have: If $\varepsilon_a = 1$ and $\varepsilon > 0$, then (a_k, w_k) converges to global minimizers for almost all initial values (a_0, w_0).*

Sketch of proof. We first prove that the algorithm converges for any $\varepsilon_a \in (0, 1]$ and small enough ε, with any initial value (a_0, w_0) such that $\|w_0\| \geq 1$ (appendix Lemma B.12). Next, we observe that the sequence $\{\|w_k\|\}$ is monotone increasing, and thus either converges to a finite limit or diverges. The scaling property is then used to exclude the divergent case – if $\{\|w_k\|\}$ diverges, then at some k the norm $\|w_k\|$ should be large enough, and by the scaling property, it is equivalent to a case where $\|w_k\| = 1$ and ε is small, which we have proved converges. This shows that $\|w_k\|$ converges to a finite limit, from which the convergence of w_k and the loss function value can be established, after some work. This proof is fully presented in appendix Theorem B.16 and the preceding lemmas. Lastly, using the "strict saddle point" arguments (Lee et al., 2016; Panageas & Piliouras, 2017), we can prove the set of initial value for which (a_k, w_k) converges to saddle points has Lebesgue measure 0, provided $\varepsilon_a = 1, \varepsilon > 0$ (appendix Lemma B.19). □

It is important to note that BNGD converges for all step size $\varepsilon > 0$ of w_k, independent of the spectral properties of H. This is a significant advantage and is in stark contrast with GD, where the step size is limited by $2/\lambda_{\max}$, and the condition number of H intimately controls the stability and convergence rate. Although we only prove the almost everywhere convergence to a global minimizer for the case of $\varepsilon_a = 1$, we have not encountered convergence to saddles in the OLS experiments even for $\varepsilon_a \in (0, 2)$ with initial values (a_0, w_0) drawn from typical distributions.

Remark: In appendix A, we show that the combination of the scaling property and the monotonicity of weight norms, which hold for batch (and weight) normalization of general loss functions, can be used to prove a more general convergence result: if iterates converge for small enough ε, then gradient norm converges for any ε. We note that in the independent work of Arora et al. (2019), similar ideas have been used to prove convergence results for batch normalization for neural networks. Lastly, one can also show that in the general case, the over-parameterization due to batch (and weight) normalization only introduces strict saddle points (see appendix Lemma A.1).

3.3. Convergence Rate and Acceleration Due to Over-parameterization

Having established the convergence of BNGD on OLS, a natural follow-up question is why should one use BNGD over GD. After all, even if BNGD converges for any learning rate, if the convergence is universally slower than GD then it does not offer any advantages. We prove the following result that shows that under mild conditions, the convergence rate of BNGD on OLS is linear. Moreover, close to the optima the linear rate of convergence can be shown to be faster than the best-case linear convergence rate of GD. This offers a concrete result that shows that BNGD could out-perform GD, even if the latter is perfectly-tuned.

Theorem 3.4 (Convergence rate)**.** *If (a_k, w_k) converges to a minimizer with $\hat{\varepsilon} := \lim_{k\to\infty} \hat{\varepsilon}_k < \varepsilon^*_{max} := 2/\lambda^*_{max}$, then the convergence is linear. Furthermore, when (a_k, w_k) is close to a minimizer, such that $\frac{\lambda_{max}\varepsilon|a_k|}{\sigma_k^2}\|e_k\|_H \leq \delta < 1$ (this must happen for large enough k, since we assumed convergence to a minimizer), then we have*

$$\|e_{k+1}\|_H \leq \tfrac{\rho^*(I-\hat{\varepsilon}_k H^*)+\delta}{1-\delta}\|e_k\|_H, \tag{13}$$

where $\rho^*(I-\hat{\varepsilon}_k H) := \max\{|1-\hat{\varepsilon}_k\lambda^*_{min}|, |1-\hat{\varepsilon}_k\lambda^*_{max}|\}$.

This statement is proved in appendix Lemma B.21. Recall that H^*, λ^*_{max} are defined in section 2. The assumption $\hat{\varepsilon} < \varepsilon^*_{max}$ is mild since one can prove the set of initial values (a_0, w_0) such that (a_k, w_k) converges to a minimizer (a^*, w^*) with $\hat{\varepsilon} > \varepsilon^*_{max}$ and $\det(I - \hat{\varepsilon}H^*) \neq 0$ is of measure zero (see appendix Lemma B.23).

The inequality (13) is motivated by the linearized system corresponding to Eq. (7)-(8) near a minimizer. When the iteration converges to a minimizer, the limiting $\hat{\varepsilon}$ must be a positive number where the assumption $\hat{\varepsilon} < \varepsilon^*_{max}$ makes sure the coefficient in Eq. (13) is smaller than 1. This implies linear convergence of $\|e_k\|_H$. Generally, the matrix H^* has better spectral properties than H, in the sense that $\rho^*(I - \hat{\varepsilon}_k H^*) \leq \rho(I - \hat{\varepsilon}_k H)$, provided $\hat{\varepsilon}_k > 0$, where the inequality is strict for almost all u. This is a consequence of the Cauchy eigenvalue interlacing property, which one can show directly using mini-max properties of eigenvalues (see appendix Lemma B.1). This leads to acceleration effects of BNGD: When $\|e_k\|_H$ is small, the contraction coefficient ρ in Eq. (12) can be improved to a lower coefficient in Eq. (13). This acceleration could be significant when κ^* is much smaller than κ, which can happen if the spectral gap of H is very large.

The acceleration effect can be understood heuristically as follows: due to the over-parameterization introduced by BN, the convergence rate near a minimizer is governed by H^* instead of H. The former has a degenerate direction $\{\lambda u : \lambda \in \mathbb{R}\}$, which coincides with the degenerate global minima. Hence, the effective condition number governing convergence is dependent on the largest and the second smallest eigenvalue of H^* (the smallest being 0 in the degenerate minima direction). One can contrast this with the GD case where the smallest eigenvalue of H is considered instead since no degenerate directions exists.

3.4. Robustness and Acceleration Due to Learning Rate Insensitivity

Let us now discuss another advantage BNGD possesses over GD, related to the insensitive dependence of the effective learning rate $\hat{\varepsilon}_k$ (and by extension, the effective convergence rate in Eq. (12) or Eq. (13)) on ε. The explicit dependence of $\hat{\varepsilon}_k$ on ε is quite complex, but we can give the following asymptotic estimates (see appendix B.6 for proof).

Proposition 3.5. *Suppose $\varepsilon_a \in (0, 1], a_0 w_0^T g > 0$, and $||g||^2 \geq \frac{w_0^T g}{\sigma_0^2} g^T H w_0$, then*

(1) When ε is small enough, $\varepsilon \ll 1$, the effective step size $\hat{\varepsilon}_k$ has a same order with ε.

(2) When ε is large enough, $\varepsilon \gg 1$, the effective step size $\hat{\varepsilon}_k$ has order $O(\varepsilon^{-1})$.

Observe that for finite k, $\hat{\varepsilon}_k$ is a differentiable function of ε. Therefore, the above result implies, via the mean value theorem, the existence of some $\varepsilon_0 > 0$ such that $d\hat{\varepsilon}_k/d\varepsilon|_{\varepsilon=\varepsilon_0} = 0$. Consequently, there is at least some small interval of the choice of learning rates ε where the performance of BNGD is insensitive to this choice.

In fact, empirically this is one commonly observed advantage of BNGD over GD, where the former typically allows for a variety of (large) learning rates to be used without adversely affecting performance. The same is not true for GD, where the convergence rate depends sensitively on the choice of learning rate. We will see later in Section 4 that although we only have a local insensitivity result above, the interval of this insensitivity is actually quite large in practice.

Furthermore, with some additional assumptions and approximations, the explicit dependence of $\hat{\varepsilon}_k$ on ϵ can be characterized in a quantitative manner. Concretely, we quantify the insensitivity of step size characterized by the interval in which the $\hat{\varepsilon}$ is close to the optimal step size ε_{opt} (or the maximal allowed step size ε_{max} in GD, since ε_{opt} is very close to ε_{max} when κ is large). Proposition 3.5 indicates that this interval is approximately $[C_1\varepsilon_{max}, \frac{C_2}{\varepsilon_{max}}]$, which crosses a magnitude of $\frac{C_2}{C_1\varepsilon_{max}^2}$, where C_1, C_2 are positive constants.

We set $\varepsilon_a = 1, a_0 = w_0^T g/\sigma_0$ (which is the value in the second step if we set $a_0 = 0$), $\|w_0\| = \|u\| = 1$, where Theorem 3.3 gives the linear converge result for almost all initial values and the convergence rate can be quantified by the limiting effective learning rate $\hat{\varepsilon} := \lim_{k\to\infty} \hat{\varepsilon}_k = \frac{\varepsilon}{\|w_\infty\|^2}$. Consequently, we need to estimate the magnitude $\|w_\infty\|^2$. The BNGD iteration implies the following equality,

$$\|w_{k+1}\|^2 = \|w_k\|^2 + \tfrac{\varepsilon^2}{\|w_k\|^2}\beta_k, \tag{14}$$

where β_k is defined as $\beta_k := \frac{a_k^2\|w_k\|^2}{\sigma_k^2}\|e_k\|_{H^2}^2$. The earlier convergence results motivate the following plausible approximation: we assume β_k linearly converges to zero and the iteration of $\|w_k\|^2$ can be approximated by $\xi(k+1)$ which obeys the following ODE (whose discretization formally matches Eq. (14), assuming the aforementioned convergence rate ρ):

$$\xi(0) = \|w_1\|^2, \qquad \dot{\xi}(t) = \tfrac{\varepsilon^2\beta_0\rho^{2t}}{\xi(t)}. \tag{15}$$

Its solution is $\xi^2(t) = \xi^2(0) + \frac{\varepsilon^2\beta_0}{|\ln\rho|}(1-\rho^{2t})$, where $\rho \in (0,1)$ depends on ε and is self-consistently determined by the limiting effective step size, i.e. ρ is the spectral radius of $I - \frac{\varepsilon}{\xi(\infty)}H$ and $\xi(\infty)$ in turn depends on ρ. Analyzing the dependence of $\xi(\infty)$ on ε can give an estimate of the insensitivity interval, which is now $[\varepsilon_{max}, \frac{1}{\beta_0\varepsilon_{max}}]$, since $\hat{\varepsilon} \approx \varepsilon$ when $\varepsilon \ll 1$, and $\hat{\varepsilon} \approx \frac{1}{\beta_0\varepsilon}$ when $\varepsilon \gg 1$. (see appendix B.6.) Therefore, the magnitude of the interval of insensitivity varies inversely with β_0. Below, we quantify this magnitude in an average sense.

Definition 3.6. *The average magnitude of the insensitivity interval of BNGD with $\varepsilon_a = 1, a_0 = \frac{w_0^T g}{\sigma_0}$ (or $a_0 = 0$) is defined as $\Omega_H = 1/(\bar{\beta}_H\varepsilon_{max}^2)$, where $\bar{\beta}_H$ is the geometric average of β_0 over w_0 and u, which we take to be independent and uniformly on the unit sphere $\mathbb{S}^{d-1}$,*

$$\bar{\beta}_H := \exp\left(\mathbb{E}_{w_0,u}\ln\left[\left(\tfrac{w_0^T Hu}{w_0^T Hw_0}\right)^2\|e_0\|_{H^2}^2\right]\right). \tag{16}$$

Note that we use the geometric average rather than the arithmetic average because we are measuring a ratio. Although we can not calculate the value of Ω_H analytically, we have the following lower bound (see appendix B.7):

Proposition 3.7. *For positive definite matrix H with minimal and maximal eigenvalues λ_{min} and λ_{max} respectively, the Ω_H defined in Definition 3.6 satisfies $\Omega_H \geq \frac{d}{C}$, where*

$$C := 4\tfrac{Tr[H^2]}{d\lambda_{min}^2}\tfrac{Tr[H]}{d\lambda_{max}}\exp\left(\tfrac{2\ln\kappa}{\kappa-1}\left(1-\tfrac{Tr[H]}{d\lambda_{min}}\right)\right), \tag{17}$$

$\kappa = \frac{\lambda_{max}}{\lambda_{min}}$ is the condition number of H.

As a consequence of the above, if the eigenvalues of H are sampled from a given continuous distribution on $[\lambda_{min}, \lambda_{max}]$, such as the uniform distribution, then by law of large numbers, $\Omega_H = O(d)$ for large d. This result suggests that the magnitude of the interval on which the performance of BNGD is insensitive to the choice of the learning rate increases linearly in dimension, implying that this robustness effect of BNGD is especially useful for high dimensional problems. Interestingly, although we only derived this result for the OLS problem, this linear scaling of insensitivity interval is also observed in neural networks experiments, where we varied the dimension by adjusting the width of the hidden layers. See Section 4.2.

The insensitivity to learning rate choices can also lead to acceleration effects if one have to use the same learning rate for training weights with different effective conditioning. This may arise in deep learning applications where each layer's gradient magnitude varies widely, thus requiring very different learning rates to achieve good performance. In this case, BNGD's large range of learning rate insensitivity allows one to use common values across all layers without adversely affecting the performance. This is again in contrast to GD, where such insensitivity is not present. See Section 4.3 for some experimental validation of this claim.

4. Experiments

Let us first summarize our key findings and insights from the analysis of BNGD on the OLS problem.

1. A scaling law governs BNGD, where certain configurations can be deemed equivalent.
2. BNGD converges for any learning rate $\varepsilon > 0$, provided that $\varepsilon_a \in (0, 1]$. In particular, different learning rates can be used for the BN variables (a) compared with the remaining trainable variables (w).
3. There exists intervals of ε for which the performance of BNGD is not sensitive to the choice of ε, and the magnitude of this interval grows with dimension.

In the subsequent sections, we first validate numerically these claims on the OLS model, and then show that these insights go beyond the simple OLS model we considered in the theoretical framework. In fact, much of the uncovered properties are observed in general applications of BNGD in deep learning.

4.1. Experiments on OLS

Here we test the convergence and stability of BNGD for the OLS model. Consider a diagonal matrix $H = \text{diag}(h)$ where $h = (1, ..., \kappa)$ is a increasing sequence. The scaling property (Proposition 3.2) allows us to set the initial value w_0 having same norm with u, $\|w_0\| = \|u\| = 1$. Of course, one can verify that the scaling property holds strictly in this case.

Figure 1 gives examples of H with different condition numbers κ. We tested the loss function of BNGD, compared with the optimal GD (i.e. GD with the optimal step size ε_{opt}), in a large range of step sizes ε_a and ε, and with different initial values of a_0. Another quantity we observe is the effective step size $\hat{\varepsilon}_k$ of BN. The results are encoded by four different colors: whether $\hat{\varepsilon}_k$ is close to the optimal step size ε_{opt}, and whether loss of BNGD is less than the optimal GD. The results indicate that the optimal convergence rate of BNGD can be better than GD in some configurations, consistent with the statement of Theorem 3.4. Recall that this acceleration phenomenon is ascribed to the conditioning of H^* which is better than H.

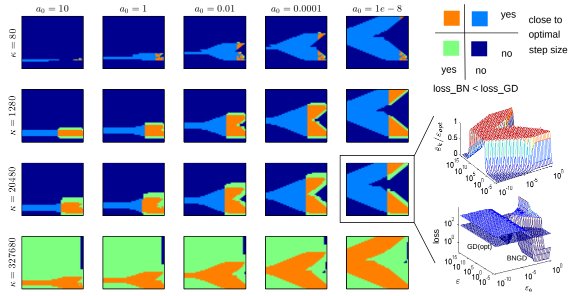

Figure 1. Comparison of BNGD and GD on OLS model. The results are encoded by four different colors: whether $\hat{\varepsilon}_k$ is close to the optimal step size ε_{opt} of GD, characterized by the inequality $0.8\varepsilon_{opt} < \hat{\varepsilon}_k < \varepsilon_{opt}/0.8$, and whether loss of BNGD is less than the optimal GD. Parameters: H = diag(logspace(0,log10(κ),100)), u is randomly chosen uniformly from the unit sphere in $\mathbb{R}^{100}$, w_0 is set to $Hu/\|Hu\|$. The GD and BNGD iterations are executed for $k = 2000$ steps with the same w_0. In each image, the range of ε_a (x-axis) is 1.99 * logspace(-10,0,41), and the range of ε (y-axis) is logspace(-5,16,43). Observe that the performance of BNGD is less sensitive to the condition number, and its advantage is more pronounced when the latter is big.

Another important observation is a region such that $\hat{\varepsilon}$ is close to ε_{opt}, in other words, BNGD significantly extends the range of "optimal" step sizes. Consequently, we can choose step sizes in BNGD at greater liberty to obtain almost the same or better convergence rate than the optimal GD. However, the size of this region is inversely dependent on the initial condition a_0. Hence, this suggests that small a_0 at first steps may improve robustness. On the other hand, small ε_a will weaken the performance of BN. The phenomenon suggests that improper initialization of the BN parameters weakens the power of BN. This experience is encountered in practice, such as (Cooijmans et al., 2016), where higher initial values of BN parameter are detrimental to the optimization of RNN models.

4.2. Experiments on the Effect of Dimension

In order to validate the approximate results in Section 3.4, we compute numerically the dependence of the performance of BNGD on the choice of the learning rate ε. Observe from Figure 2 that the quantitative predictions of Ω in Definition 3.6 is consistent with numerical experiments, and the linear-in-dimension scaling of the magnitude of the insensitivity interval is observed. Perhaps more interestingly, the same scaling is also observed in (stochastic) BNGD on fully connected neural networks trained on the MNIST dataset. This suggests that this scaling is relevant, at least qualitatively, beyond the regimes considered in the theoretical parts of this paper.

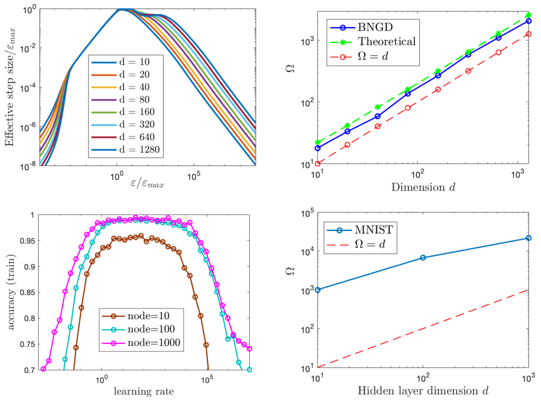

Figure 2. Effect of dimension. (Top line) Tests of BNGD on OLS model with step size $\varepsilon_a = 1, a_0 = 0$. Parameters: H=diag(linspace(1,10000,d)), u and w_0 is randomly chosen uniformly from the unit sphere in $\mathbb{R}^d$. The BNGD iterations are executed for $k = 5000$ steps. The values are averaged over 500 independent runs. (Bottom line) Tests of stochastic BNGD on MNIST dataset, fully connected neural network with one hidden layer and softmax mean-square loss. The separated learning rate for BN Parameters is lr_a=10, The performance is characterized by the accuracy at the first epoch (averaged over 10 independent runs). The magnitude Ω is approximately measured for reference.

4.3. Further Neural Network Experiments

We conduct further experiments on deep learning applied to standard classification datasets: MNIST (LeCun et al., 1998), Fashion MNIST (Xiao et al., 2017) and CIFAR-10 (Krizhevsky & Hinton, 2009). The goal is to explore if the other key findings outlined at the beginning of this section continue to hold for more general settings. For the MNIST and Fashion MNIST dataset, we use two different networks: (1) a one-layer fully connected network (784 × 10) with softmax mean-square loss; (2) a four-layer convolution network (Conv-MaxPool-Conv-MaxPool-FC-FC) with ReLU activation function and cross-entropy loss. For the CIFAR-10 dataset, we use a five-layer convolution network (Conv-MaxPool-Conv-MaxPool-FC-FC-FC). All the trainable parameters are randomly initialized by the Glorot scheme (Glorot & Bengio, 2010) before training. For all three datasets, we use a minibatch size of 100 for computing stochastic gradients. In the BNGD experiments, batch normalization is performed on all layers, the BN parameters are initialized to transform the input to zero mean/unit variance distributions, and a small regularization parameter $\epsilon =$1e-3 is added to variance $\sqrt{\sigma^2+\epsilon}$ to avoid division by zero.

Scaling property Theoretically, the scaling property 3.2 holds for any layer using BN. However, it may be slightly biased by the regularization parameter ϵ. Here, we test the scaling property in practical settings. Figure 3 gives the loss and accuracy of network-(2) (2CNN+2FC) at the first epoch with different learning rate. The norm of all weights and biases are rescaled by a common factor η. We observe that the scaling property remains true for relatively large η. However, when η is small, the norm of weights are small. Therefore, the effect of the ϵ-regularization in $\sqrt{\sigma^2+\epsilon}$ becomes significant, causing the curves to be shifted.

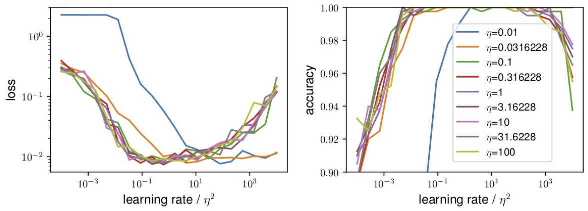

Figure 3. Tests of scaling property of the 2CNN+2FC network on MNIST dataset. BN is performed on all layers, and ϵ=1e-3 is added to variance $\sqrt{\sigma^2+\epsilon}$. All the trainable parameters (except the BN parameters) are randomly initialized by the Glorot scheme, and then multiplied by a same parameter η.

Stability for large learning rates We use the loss value at the end of the first epoch to characterize the performance of BNGD and GD methods. Although the training of models have generally not converged at this point, it is enough to extract some relative rate information. Figure 4 shows the loss value of the networks on the three datasets. It is observed that GD and BNGD with identical learning rates for weights and BN parameters exhibit a maximum allowed learning rate, beyond which the iterations becomes unstable. On the other hand, BNGD with separate learning rates exhibits a much larger range of stability over learning rate for non-BN parameters, consistent with our theoretical results on OLS problem

Insensitivity of performance to learning rates Observe that BN accelerates convergence more significantly for deep networks, whereas for one-layer networks, the best performance of BNGD and GD are similar. Furthermore, in most cases, the range of optimal learning rates in BNGD is quite large, which is in agreement with the OLS analysis (see Section 3.4). This phenomenon is potentially crucial for understanding the acceleration of BNGD in deep neural networks. Heuristically, the "optimal" learning rates of GD in distinct layers (depending on some effective notion of "condition number") may be vastly different. Hence, GD with a shared learning rate across all layers may not achieve the best convergence rates for all layers at the same time. In this case, it is plausible that the acceleration of BNGD is a result of the decreased sensitivity of its convergence rate on the learning rate parameter over a large range of its choice.

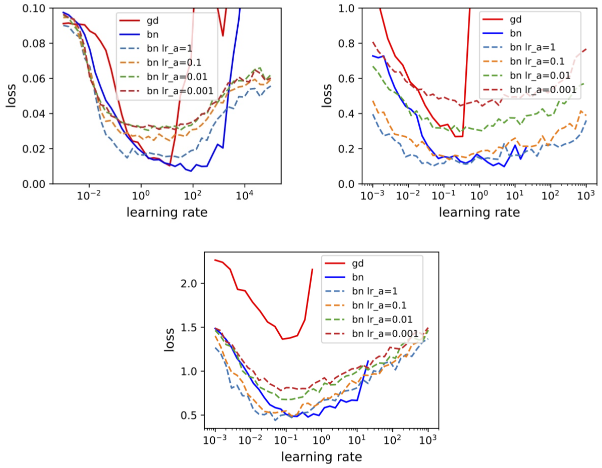

Figure 4. Performance of BNGD and GD method on MNIST (network-(1), 1FC), Fashion MNIST (network-(2), 2CNN+2FC) and CIFAR-10 (2CNN+3FC) datasets. The performance is characterized by the loss value at the first epoch. In the BNGD method, both the shared learning rate schemes and separated learning rate scheme (learning rate lr_a for BN parameters) are given. The values are averaged over 5 independent runs.

5. Conclusion

In this paper, we analyzed the dynamical properties of batch normalization on OLS, chosen for its simplicity and the availability of precise characterizations of GD dynamics. Even in such a simple setting, we saw that BNGD exhibits interesting non-trivial behavior, including scaling laws, robust convergence properties, acceleration, as well as the insensitivity of performance to the choice of learning rates. At least in the setting considered here, our analysis allows one to concretely answer the question of why BNGD can achieve better performance than GD. Although these results are derived only for the OLS model, we show via experiments that these are qualitatively, and sometimes quantitatively valid for more general scenarios. These point to promising future directions towards uncovering the dynamical effect of batch normalization in deep learning and beyond.

References

Arora, S., Li, Z., and Lyu, K. Theoretical analysis of auto rate-tuning by batch normalization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkxQ-nA9FX.

Bjorck, J., Gomes, C., and Selman, B. Understanding Batch Normalization. *ArXiv e-prints*, May 2018.

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

Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. Lower bounds for finding stationary points i. *arXiv preprint arXiv:1710.11606*, 2017.

Cooijmans, T., Ballas, N., Laurent, C., and Courville, A. C. Recurrent batch normalization. *CoRR*, abs/1603.09025, 2016. URL http://arxiv.org/abs/1603.09025.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Ioffe, S. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. *CoRR*, abs/1702.03275, 2017. URL http://arxiv.org/abs/1702.03275.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

Kohler, J., Daneshmand, H., Lucchi, A., Zhou, M., Neymeyr, K., and Hofmann, T. Towards a Theoretical Understanding of Batch Normalization. *ArXiv e-prints*, May 2018.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient Descent Converges to Minimizers. *ArXiv e-prints*, February 2016.

Ma, Y. and Klabjan, D. Convergence analysis of batch normalization for deep neural nets. *CoRR*, 1705.08011, 2017. URL http://arxiv.org/abs/1705.08011.

Panageas, I. and Piliouras, G. Gradient Descent Only Converges to Minimizers: Non-Isolated Critical Points and Invariant Regions. In Papadimitriou, C. H. (ed.), *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 2:1–2:12, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-029-3. doi: 10.4230/LIPIcs.ITCS.2017.2.

Saad, Y. *Iterative methods for sparse linear systems*, volume 82. siam, 2003.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. How Does Batch Normalization Help Optimization? (No, It Is Not About Internal Covariate Shift). *ArXiv e-prints*, May 2018.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.