Gradient Diversity: a Key Ingredient for Scalable Distributed Learning

Dong Yin
UC Berkeley

Ashwin Pananjady
UC Berkeley

Max Lam
Stanford University

Dimitris Papailiopoulos
UW-Madison

Kannan Ramchandran
UC Berkeley

Peter L. Bartlett
UC Berkeley

Abstract

It has been experimentally observed that distributed implementations of mini-batch stochastic gradient descent (SGD) algorithms exhibit speedup saturation and decaying generalization ability beyond a particular batch-size. In this work, we present an analysis hinting that high similarity between concurrently processed gradients may be a cause of this performance degradation. We introduce the notion of *gradient diversity* that measures the dissimilarity between concurrent gradient updates, and show its key role in the convergence and generalization performance of mini-batch SGD. We also establish that heuristics similar to DropConnect, Langevin dynamics, and quantization, are provably diversity-inducing mechanisms, and provide experimental evidence indicating that these mechanisms can indeed enable the use of larger batches without sacrificing accuracy and lead to faster training in distributed learning. For example, in one of our experiments, for a convolutional neural network to reach 95% training accuracy on MNIST, using the diversity-inducing mechanism can reduce the training time by 30% in the distributed setting.

1 INTRODUCTION

In recent years, deploying algorithms on distributed computing units has become the *de facto* architectural choice for large-scale machine learning. Distributed optimization has gained significant traction with a large body of recent work establishing near-optimal speedup gains on both convex and nonconvex objectives (Chen et al., 2016, Dean et al., 2012, Duchi et al., 2013, Gemulla et al., 2011, Jaggi et al., 2014, Liu et al., 2014, Niu et al., 2011, Yun et al., 2013), and several state-of-the-art publicly available (distributed) machine learning frameworks, such as Tensorflow (Abadi et al., 2016), MXNet (Chen et al.), and Caffe2 (Chilimbi et al., 2014), offer distributed implementations of popular learning algorithms.

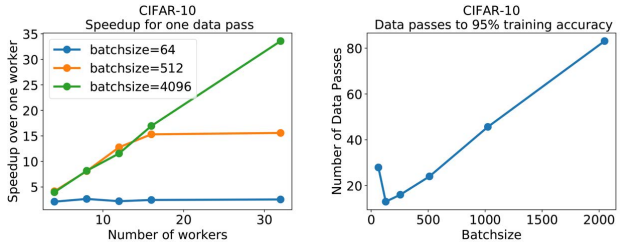
Mini-batch stochastic gradient descent (SGD) is the algorithmic cornerstone for several of these distributed frameworks. During a distributed iteration of mini-batch SGD, a master node stores a global model, and P worker nodes compute gradients for B data points, sampled from a total of n training data (*i.e.,* B/P samples per worker per iteration), with respect to the same global model; the parameter B is commonly referred to as the batch-size. The master, after receiving these B gradients, applies them to the model and sends the updated model back to the workers; this is the equivalent of one round of communication.

Unfortunately, near-optimal scaling for distributed variants of mini-batch SGD is only possible for up to tens of compute nodes. Several studies (Dean et al., 2012, Qi et al., 2016) indicate that there is a significant gap between ideal and realizable speedups when scaling out to hundreds of compute nodes. This commonly observed phenomenon is referred to as *speedup saturation*. A key cause of speedup saturation is the communication overhead of mini-batch SGD.

Ultimately, the batch-size B controls a crucial performance trade-off between communication cost and convergence speed, as observed and analyzed in several studies (Goyal et al., 2017, Takác et al., 2013, Wang et al., 2017). When using large batch-sizes, we observe large speedup gains per pass (*i.e.,* per n gradient computations), as shown in Figure 1a, due to fewer communication rounds. However, as shown in Figure 1b, to achieve a desired level of accuracy for larger batches, we may need a larger number of passes over the dataset, resulting in *overall* slower computa-

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

tion that leads to speedup saturation. Furthermore, recent work shows that large batch sizes lead to models that generalize worse (Keskar et al., 2016), and efforts have been made to improve the generalization ability (Hoffer et al., 2017).



(a) (b)

Figure 1: (a) Speedup gains for a single data pass and various batch-sizes, for a cuda-convnet model on CIFAR-10. (b) Number of data passes to reach 95% accuracy for a cuda-convnet model on CIFAR-10, vs batch-size. Step-sizes are tuned to maximize convergence speed.

The key question that motivates our work is: *How does the batch-size control the scalability and generalization performance of mini-batch SGD*?

1.1 Our Contributions

We employ the notion of *gradient diversity* that measures the dissimilarity between concurrent gradient updates. We show that the convergence of mini-batch SGD, on both convex and nonconvex loss functions, including the Polyak-Łojasiewicz functions (Lojasiewicz, 1963, Polyak, 1963), is identical—up to constant factors—to that of serial SGD (*e.g.,* $B = 1$), if the batch-size is proportional to a bound implied by gradient diversity. We also establish the worst case optimality and tightness of the bound in strongly convex functions.

Although it has been empirically observed that more diversity in the data leads to more parallelism (Chilimbi et al., 2014), and there has been significant work on the theory of mini-batch algorithms, our results have two major novelties: 1) our batch-size bound is data-dependent, tight, and essentially identical across convex and nonconvex functions, and in some cases leads to guaranteed uniformly larger batch-sizes compared to prior work, and 2) the bound has an operational meaning, and inspired by our theory, we establish that algorithmic heuristics similar to DropConnect (Wan et al., 2013), Langevin dynamics (Welling and Teh, 2011), and quantization (Alistarh et al., 2016) are diversity-inducing mechanisms. In our experiments, we find that the proposed mechanisms can indeed enable the use of larger batch-size in distributed learning, and thus reduce training time.

Following our convergence analysis, we study the effect of batch-size on the generalization behavior of mini-batch SGD using the notion of algorithmic stability (Bousquet and Elisseeff, 2002). Through a similar measure of gradient diversity, we show that as long as the batch-size is below a certain threshold, then mini-batch SGD is as stable as one sample SGD that is analyzed by Hardt et al. (2015).

2 RELATED WORK

Mini-batch SGD Dekel et al. (2012) analyze mini-batch SGD on non-strongly convex functions and propose $B = \mathcal{O}(\sqrt{T})$ as an optimal choice for batch-size. In contrast, our work provides a data-dependent principle for the choice of batch-size, and it holds without the requirement of convexity. Even in the regime where the result in Dekel et al. (2012) is valid, depending on the problem, our result may still provide better bounds on the batch-size than $\mathcal{O}(\sqrt{T})$ (*e.g.,* in the sparse conflict setting shown in Section 4.1). Friedlander and Schmidt (2012) propose an adaptive batch-size scheme and show that this scheme provides weak linear convergence rate for strongly convex functions. De et al. (2016) propose an optimization algorithm for choosing the batch-size, and weighted sampling techniques have also been developed (Needell and Ward, 2016, Zhang et al., 2017, Zhao and Zhang, 2014).

Diversity and data-dependent bounds In empirical studies, it has been observed that more diversity in the data allows more parallelism (Chilimbi et al., 2014). As for the theoretical analysis, data-dependent thresholds for batch-size have been developed for some specific problems such as least squares (Jain et al., 2016) and SVM (Takác et al., 2013). In particular, for least square problems, Jain et al. (2016) propose a bound on batch-size similar to our measure of gradient diversity; however, as mentioned in Section 1, our result holds for a wider range of problems including nonconvex setups, and can be used to motivate heuristics that result in speedup gains in distributed systems.

Other mini-batching and distributed optimization algorithms Beyond mini-batch SGD, several other mini-batching algorithms have been proposed; we survey a non-exhaustive list. Mini-batch proximal algorithms are studied by Li et al. (2014), Wang et al. (2017), Wang and Srebro (2017), and these algorithms require solving a regularized optimization algorithm on a sampled batch as a subroutine. Other algorithms include accelerated methods (Cotter et al., 2011), mini-batch SDCA (Shalev-Shwartz and Zhang, 2013, Takáč et al., 2015), and the combination of mini-batching and variance reduction such as Acc-Prox-SVRG (Nitanda, 2014) and mS2GD (Konečnỳ et al., 2016). Here, we emphasize that although different mini-batching algorithms can be designed for particular problems and may work better in particular regimes, especially in

the convex setting, these algorithms are usually more difficult to implement in distributed learning frameworks like Tensorflow or MXNet, and can introduce additional communication costs. A few other algorithms have been recently proposed to reduce the communication cost by inducing sparsity in the gradients, for instance, QSGD (Alistarh et al., 2016) and TernGrad (Wen et al., 2017).

Generalization and stability An important performance measure of a learning algorithm is its generalization ability. In their foundational work, Bousquet and Elisseeff (2002) prove the equivalence between algorithmic stability and generalization. This approach is then used to establish generalization bounds for SGD by Hardt et al. (2015). Another approach to prove generalization bounds is to use the operator view of averaged SGD (Dfossez and Bach, 2014). This method is extended by Jain et al. (2016) to the random least-squares regression problems. Variance reduction methods are also used to develop algorithms with good generalization performance (Daneshmand et al., 2016, Frostig et al., 2015). In this paper, we extend the stability approach to the mini-batch setting, and show that the generalization ability is governed by a quantity that is also function of gradient diversity.

3 PROBLEM SETUP

We consider the following general supervised learning setup. Suppose that $\mathcal{D}$ is an unknown distribution over a sample space $\mathcal{Z}$, and we have access to a sample $\mathcal{S} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ of n data points, that are drawn i.i.d. from $\mathcal{D}$. Our goal is to find a model $\mathbf{w}$ from a model space $\mathcal{W} \subseteq \mathbb{R}^d$ with small *population risk* with respect to a loss function f, *i.e.,* we want to minimize $R(\mathbf{w}) = \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}[f(\mathbf{w};\mathbf{z})]$. Since we do not have access to the population risk, we instead train a model whose aim is to minimize the *empirical risk*

$$R_{\mathcal{S}}(\mathbf{w}) := \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{w};\mathbf{z}_i). \tag{1}$$

For any training algorithm that operates on the empirical risk, there are two important aspects to analyze: the convergence speed to a good model with small empirical risk, and the generalization gap $|R_{\mathcal{S}}(\mathbf{w}) - R(\mathbf{w})|$ that quantifies the performance discrepancy of the model between the empirical and population risks. For simplicity, we use the notation $f_i(\mathbf{w}) := f(\mathbf{w};\mathbf{z}_i)$, $F(\mathbf{w}) := R_{\mathcal{S}}(\mathbf{w})$, and define $\mathbf{w}^* \in \arg\min_{\mathbf{w}\in\mathcal{W}} F(\mathbf{w})$. In this work, we focus on families of differentiable losses that satisfy a subset of the following for all parameters $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$:

Definition 1 (β-smooth)**.**
$F(\mathbf{w}) \le F(\mathbf{w}') + \langle \nabla F(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle + \frac{\beta}{2}\|\mathbf{w} - \mathbf{w}'\|_2^2$.

Definition 2 (λ-strongly convex)**.**
$F(\mathbf{w}) \ge F(\mathbf{w}') + \langle \nabla F(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle + \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}'\|_2^2$.

Definition 3 (μ-Polyak-Łojasiewicz (PL) (Lojasiewicz, 1963, Polyak, 1963))**.**
$\frac{1}{2}\|\nabla F(\mathbf{w})\|_2^2 \ge \mu(F(\mathbf{w}) - F(\mathbf{w}^*))$.

Mini-batch SGD At each iteration, mini-batch SGD computes B gradients on randomly sampled data at the most current global model. At the $(k+1)$-th distributed iteration, the model is given by

$$\mathbf{w}_{(k+1)B} = \mathbf{w}_{kB} - \gamma \sum_{\ell=kB}^{(k+1)B-1} \nabla f_{s_\ell}(\mathbf{w}_{kB}), \tag{2}$$

where each index s_i is drawn uniformly at random from $[n]$, with replacement. Here, we use $\mathbf{w}$ with subscript kB to denote the model we obtain after k distributed iterations, *i.e.,* a total of kB gradient updates. Note that we recover serial SGD when $B = 1$. Our results also apply to varying step-size, but for simplicity we only state our bounds with constant step-size. In related work, there is a normalization of $1/B$ included in the gradient step, here, *without loss of generality* we subsume that in the step-size γ.

We note that some of our analyses require $\mathcal{W}$ to be a bounded convex subset of $\mathbb{R}^d$, where the projected version of SGD can be used, by making Euclidean projections back to $\mathcal{W}$, *i.e.,* $\mathbf{w}_{(k+1)B} = \Pi_{\mathcal{W}}(\mathbf{w}_{kB} - \gamma\sum_{\ell=kB}^{(k+1)B-1}\nabla f_{s_\ell}(\mathbf{w}_{kB}))$. For simplicity, in our main text, we refer to both with/without projection algorithms as "mini-batch SGD", but in our supplementary material we make the distinction clear when needed.

4 GRADIENT DIVERSITY AND CONVERGENCE

In this section, we introduce our definition of gradient diversity, and state our convergence results.

4.1 Gradient Diversity

Gradient Diversity quantifies the degree to which individual gradients of the loss functions are different from each other. We note that a similar notion was introduced by Jain et al. (2016) for least squares problems.

Definition 4 (gradient diversity)**.** *We refer to the following ratio as gradient diversity:*

$$\Delta_D(\mathbf{w}) := \frac{\sum_{i=1}^n \|\nabla f_i(\mathbf{w})\|_2^2}{\|\sum_{i=1}^n \nabla f_i(\mathbf{w})\|_2^2} = \frac{\sum_{i=1}^n \|\nabla f_i(\mathbf{w})\|_2^2}{\sum_{i=1}^n \|\nabla f_i(\mathbf{w})\|_2^2 + \sum_{i\neq j}\langle \nabla f_i(\mathbf{w}), \nabla f_j(\mathbf{w})\rangle}.$$

Clearly, $\Delta_D(\mathbf{w})$ is large when the inner products between the gradients taken with respect to different data points are small, and so measures diverse these gradients are. We further define a batch-size bound $B_D(\mathbf{w})$ for each data set $\mathcal{S}$ and each $\mathbf{w} \in \mathcal{W}$:

Definition 5 (batch-size bound)**.**

$$B_D(\mathbf{w}) := n\Delta_D(\mathbf{w}).$$

As we see in later parts, the batch-size bound $B_D(\mathbf{w})$ implied by gradient diversity plays a fundamental role in the batch-size selection for mini-batch SGD.

Examples of gradient diversity We provide two examples in which we can compute a uniform lower bound for all $B_D(\mathbf{w})$, $\mathbf{w} \in \mathcal{W}$. Notice that these bounds solely depend on the data set $\mathcal{S}$, and are thus *data dependent*.

Example 1 (generalized linear function) Suppose that any data point $\mathbf{z}$ consists of feature vector $\mathbf{x} \in \mathbb{R}^d$ and some label $y \in \mathbb{R}$, and for sample $\mathcal{S} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$, the loss function $f(\mathbf{w}; \mathbf{z}_i)$ can be written as a generalized linear function $f(\mathbf{w}; \mathbf{z}_i) = \ell_i(\mathbf{x}_i^{\mathrm{T}}\mathbf{w})$, where $\ell_i : \mathbb{R} \to \mathbb{R}$ is a differentiable one-dimensional function, and we do not require the convexity of $\ell_i(\cdot)$. Let $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_n]^{\mathrm{T}} \in \mathbb{R}^{n\times d}$ be the feature matrix. We have the following results for $B_D(\mathbf{w})$ for generalized linear functions.

Remark 1. *For generalized linear functions,* $\forall\ \mathbf{w} \in \mathcal{W}$, $B_D(\mathbf{w}) \geq n \min_{i=1,\ldots,n} \|\mathbf{x}_i\|_2^2/\sigma_{\max}^2(\mathbf{X})$.

We note that it has been shown by Takác et al. (2013) that the spectral norm of the data matrix is important for the batch-size choice for SVM problems, and our results have similar implication. In addition, suppose that $n \geq d$, and $\mathbf{x}_i$ has i.i.d. σ-sub-Gaussian entries with zero mean. Then there exist universal constants $c_1, c_2, c_3 > 0$ such that with probability $1 - c_2 n e^{-c_3 d}$, $B_D(\mathbf{w}) \geq c_1 d$, $\forall\ \mathbf{w} \in \mathcal{W}$. Therefore, as long as we are in the relatively high dimensional regime $d = \Omega(\log(n))$, we have $B_D(\mathbf{w}) \geq \Omega(d)$, $\forall\ \mathbf{w} \in \mathcal{W}$ with high probability.

Example 2 (sparse conflicts) In some applications (Joachims, 2006), the gradient of an individual loss function $\nabla f_i(\mathbf{w})$ depends only on a small subset of all the coordinates of $\mathbf{w}$ (called the support), and the supports of the gradients have *sparse conflict*. More specifically, define a graph $G = (V, E)$ with the vertices V representing the n data points, and $(i, j) \in E$ when the supports of $\nabla f_i(\mathbf{w})$ and $\nabla f_j(\mathbf{w})$ have non-empty overlap. Let ρ be the maximum degree of all the vertices in G.

Remark 2. *For sparse conflicts, we have* $B_D(\mathbf{w}) \geq n/(\rho+1)$ *for all* $\mathbf{w} \in \mathcal{W}$. *This bound can be large when* G *is sparse,* i.e., *when* ρ *is small.*

4.2 Convergence Rates

Our convergence results are consequences of the following lemma, which does not require convexity of the losses, and captures the effect of mini-batching on an iterate-by-iterate basis. Here, we define $M^2(\mathbf{w}) := \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(\mathbf{w})\|_2^2$ for any $\mathbf{w} \in \mathcal{W}$.

Lemma 1. *Let* $\mathbf{w}_{kB}$ *be a fixed model, and let* $\mathbf{w}_{(k+1)B}$ *denote the model after a mini-batch iteration with batch-size* $B = \delta B_D(\mathbf{w}_{kB}) + 1$. *Then*

$$\begin{aligned}\mathbb{E}[\|\mathbf{w}_{(k+1)B} - \mathbf{w}^*\|_2^2 \mid \mathbf{w}_{kB}] \\ \leq \|\mathbf{w}_{kB} - \mathbf{w}^*\|_2^2 - 2B\gamma\langle\nabla F(\mathbf{w}_{kB}), \mathbf{w}_{kB} - \mathbf{w}^*\rangle \\ + (1+\delta)B\gamma^2 M^2(\mathbf{w}_{kB}),\end{aligned}$$

where equality holds when there are no projections.

As one can see, for a single iteration, in expectation, the model trained by serial SGD ($B = 1$), closes the distance to the optimal by exactly $2\gamma\langle\nabla F(\mathbf{w}_{kB}), \mathbf{w}_{kB} - \mathbf{w}^*\rangle - \gamma^2 M^2(\mathbf{w}_{kB})$. Our bound says that using the *same* step-size[1] as SGD (*without normalizing* with a factor of B), mini-batch will close that distance to the optimal (or any critical point $\mathbf{w}^*$) by approximately B times more, if $B = \mathcal{O}(B_D(\mathbf{w}_{kB}))$. This matches the best that we could have hoped for: mini-batch SGD with batch-size B should be B times faster per iteration than a single iteration of serial SGD.

We now provide convergence results using gradient diversity. For a mini-batch SGD algorithm, define the set $\mathcal{W}_T \subset \mathcal{W}$ as the collection of all possible model parameters that the algorithm can reach during T/B parallel iterations, *i.e.,*

$$\mathcal{W}_T := \{\mathbf{w} \in \mathcal{W} \ :\ \mathbf{w} = \mathbf{w}_{kB} \text{ for some instance of mini-batch SGD, } k = 0, 1, \ldots, T/B\}.$$

Our main message can be summarized as follows:

Theorem 1 (informal convergence result)**.** *Let* $B \leq \delta B_D(\mathbf{w}) + 1$, $\forall\ \mathbf{w} \in \mathcal{W}_T$. *If serial SGD achieves an* ϵ*-suboptimal*[2] *solution after* T *gradient updates, then using the same step-size as serial SGD, mini-batch SGD with batch-size* B *can achieve a* $(1+\frac{\delta}{2})\epsilon$*-suboptimal solution after the same number of gradient updates (*i.e., T/B *iterations).*

Therefore, our result implies that, as long as the batch-size does not exceed the fundamental bound implied by gradient diversity, using the *same* step-size as the serial algorithm, mini-batch SGD does not suffer from convergence speed saturation.

We provide the precise statements of the results as follows. Define $F^* = \min_{\mathbf{w}\in\mathcal{W}} F(\mathbf{w})$, $D_0 = \|\mathbf{w}_0 - \mathbf{w}^*\|_2^2$. In all the following results, we assume that $B \leq \delta B_D(\mathbf{w}) + 1$, $\forall\ \mathbf{w} \in \mathcal{W}_T$, and $M^2(\mathbf{w}) \leq M^2$, $\forall\ \mathbf{w} \in \mathcal{W}_T$. The step-sizes in the following results are known to be the order-optimal choices for serial SGD with constant step-size (Bottou et al., 2016, Ghadimi and Lan, 2016, Karimi et al., 2016). We start with more general function classes, *i.e.,* nonconvex smooth functions and PL functions.

[1] In fact, our choice of step-size is consistent with many state-of-the-art distributed learning frameworks (Goyal et al., 2017), and we would like to point out that our paper provides theoretical explanation of this choice of step-size.

[2] Suboptimality is defined differently for different classes of functions.

Theorem 2 (smooth functions). *Suppose that $F(\mathbf{w})$ is β-smooth, $\mathcal{W} = \mathbb{R}^d$, and use step-size $\gamma = \frac{\epsilon}{\beta M^2}$. Then, after $T \geq \frac{2}{\epsilon^2} M^2 \beta (F(\mathbf{w}_0) - F^*)$ gradient updates, $\min_{k=0,\ldots,T/B-1} \mathbb{E}[\|\nabla F(\mathbf{w}_{kB})\|_2^2] \leq (1 + \frac{\delta}{2})\epsilon$.*

Theorem 3 (PL functions). *Suppose that $F(\mathbf{w})$ is β-smooth, μ-PL, $\mathcal{W} = \mathbb{R}^d$, and use step-size $\gamma = \frac{2\epsilon\mu}{M^2\beta}$, and batch-size $B \leq \frac{1}{2\gamma\mu}$. Then, after $T \geq \frac{M^2\beta}{4\mu^2\epsilon} \log(\frac{2(F(\mathbf{w}_0)-F^*)}{\epsilon})$ gradient updates, we have $\mathbb{E}[F(\mathbf{w}_T) - F^*] \leq (1 + \frac{\delta}{2})\epsilon$.*

For convex loss functions, we emphasize that, there have been a lot of studies that establish similar rates, without explicitly using our notion of gradient diversity (Friedlander and Schmidt, 2012, Jain et al., 2016, Takác et al., 2013). We present the results for completeness, and also note that via gradient diversity, we provide a general form of convergence rates that is essentially identical across convex and nonconvex objectives.

Theorem 4 (convex functions). *Suppose that $F(\mathbf{w})$ is convex, and use step-size $\gamma = \frac{\epsilon}{M^2}$. Then, after $T \geq \frac{M^2 D_0}{\epsilon^2}$ gradient updates, we have $\mathbb{E}[F(\frac{B}{T}\sum_{k=0}^{\frac{T}{B}-1} \mathbf{w}_{kB}) - F^*] \leq (1 + \frac{\delta}{2})\epsilon$.*

Theorem 5 (strongly convex functions). *Suppose that $F(\mathbf{w})$ is λ-strongly convex, and use step-size $\gamma = \frac{\epsilon\lambda}{M^2}$ and batch-size $B \leq \frac{1}{2\lambda\gamma}$. Then, after $T \geq \frac{M^2}{2\lambda^2\epsilon} \log(\frac{2D_0}{\epsilon})$ gradient updates, we have $\mathbb{E}[\|\mathbf{w}_T - \mathbf{w}^*\|_2^2] \leq (1 + \frac{\delta}{2})\epsilon$.*

4.3 Worst-case Optimality

Here, we establish that the above bound on the batch-size is worst-case optimal. The following theorem demonstrates this for a convex problem with varying agnostic batch-sizes[3] B_k. Essentially, if we violate the batch bound prescribed above by a factor of δ, then the quality of our model will be penalized by a factor of δ, in terms of accuracy.

Theorem 6. *Consider a mini-batch SGD algorithm with K iterations and varying batch-sizes $B_1, B_2, \ldots, B_K$, and let $N_k = \sum_{i=1}^{k} B_i$. Then, there exists a λ-strongly convex function $F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w})$ with bounded parameter space $\mathcal{W}$, such that, if $B_k \leq \frac{1}{2\lambda\gamma}$ and $B_k \geq \delta\mathbb{E}[B_D(\mathbf{w}_{N_{k-1}})] + 1 \;\forall\; k = 1, \ldots, K$ (where the expectation is taken over the randomness of the mini-batch SGD algorithm), and the total number of gradient updates $T = N_K \geq \frac{c}{\lambda\gamma}$ for some universal constant $c > 0$, we have: $\mathbb{E}[\|\mathbf{w}_T - \mathbf{w}^*\|_2^2] \geq c'(1 + \delta)\frac{\gamma M^2}{\lambda}$, where $c' > 0$ is another universal constant. More concretely, when running mini-batch SGD with step-size $\gamma = \frac{\epsilon\lambda}{M^2}$ and at least $\mathcal{O}(\frac{M^2}{\lambda^2\epsilon})$ gradient updates, we have $\mathbb{E}[\|\mathbf{w}_T - \mathbf{w}^*\|_2^2] \geq c'(1 + \delta)\epsilon$.*

Although the above bound is only for strongly convex functions, it reveals that there exist regimes beyond which scaling the batch-size beyond our fundamental bound can lead to only worse performance in terms of the accuracy for a given iteration, or the number of iterations needed for a specific accuracy.

4.4 Diversity-inducing Mechanisms

In recent years, several algorithmic heuristics, such as DropConnect (Wan et al., 2013), stochastic gradient Langevin dynamics (SGLD) (Welling and Teh, 2011), and quantization (Alistarh et al., 2016), have been shown to be useful for improving large scale optimization in various aspects. For example, they may help improve generalization or escape saddle points (Ge et al., 2015). In this section, we demonstrate a *different aspect* of these heuristics. We show that gradient diversity can increase when applying these techniques independently to the data points in a batch, rendering mini-batch SGD more amenable to distributed speedup gains.

We note that these mechanisms have two opposing effects: on one hand, as we show in the sequel they allow the use of larger batch-sizes, and thus can reduce communication cost by reducing the number of iterations; on the other hand, these methods usually introduce additional variance to the stochastic gradients, and may require more iteration to achieve a particular accuracy. Consequently, there is a communication-computation trade-off inherent to these mechanisms. By carefully exploiting this trade-off, our goal would be to see a gain in the *overall* run time. In Section 6, we provide experimental evidence to show that this run time gain can indeed be observed in real distributed systems.

We use abbreviation DIM for any diversity-inducing mechanism. When data point i is sampled, instead of making gradient update $\nabla f_i(\mathbf{w})$, the algorithm updates with a random surrogate vector $\mathbf{g}_i^{\mathsf{DIM}}(\mathbf{w})$ by introducing some additional randomness, which is acquired i.i.d. across data points and iterations.

We can thus define the corresponding gradient diversity and batch-size bounds

$$\Delta_D^{\mathsf{DIM}}(\mathbf{w}) := \frac{\sum_{i=1}^{n} \mathbb{E}\|\mathbf{g}_i^{\mathsf{DIM}}(\mathbf{w})\|_2^2}{\mathbb{E}\|\sum_{i=1}^{n} \mathbf{g}_i^{\mathsf{DIM}}(\mathbf{w})\|_2^2},$$
$$B_D^{\mathsf{DIM}}(\mathbf{w}) := n\Delta_D^{\mathsf{DIM}}(\mathbf{w}),$$

where the expectation is taken over the randomness of the mechanism. In the following parts, we first demonstrate various diversity-inducing mechanisms, and then compare $B_D^{\mathsf{DIM}}(\mathbf{w})$ with $B_D(\mathbf{w})$.

DropConnect We interpret DropConnect as updating a randomly chosen subset of all the coordinates

[3] Here, by saying that the batch-sizes are *agnostic*, we emphasize the fact that the batch-sizes are constants that are picked up without looking at the progress of the algorithm.

of the model parameter vector[4]. Let $\mathbf{D}_1, \ldots, \mathbf{D}_n$ be i.i.d. diagonal matrices with diagonal entries being i.i.d. Bernoulli random variables, and each diagonal entry is 0 with drop probability $p \in (0,1)$. When data point $\mathbf{z}_i$ is chosen, we make DropConnect update $\mathbf{g}_i^{\mathsf{drop}}(\mathbf{w}) = \mathbf{D}_i \nabla f_i(\mathbf{w})$.

Stochastic gradient Langevin dynamics (SGLD) SGLD takes the gradient updates: $\mathbf{g}_i^{\mathsf{sgld}}(\mathbf{w}) = \nabla f_i(\mathbf{w}) + \xi_i$ where ξ_i, $i = 1, \ldots, n$, are independent isotropic Gaussian noise $\mathcal{N}(0, \sigma^2 \mathbf{I})$.

Quantized gradients Define $Q(\mathbf{v})$ as the quantized version of a vector $\mathbf{v}$. More precisely, $[Q(\mathbf{v})]_\ell = \|\mathbf{v}\|_2 \operatorname{sign}(v_\ell) \eta_\ell(\mathbf{v})$, where $\eta_\ell(\mathbf{v})$'s are independent Bernoulli random variables with $\mathbb{P}\{\eta_\ell = 1\} = |v_\ell|/\|\mathbf{v}\|_2$. We let $\mathbf{g}_i^{\mathsf{quant}}(\mathbf{w}) = Q(\nabla f_i(\mathbf{w}))$.

We can show that these mechanisms increases gradient diversity, as long as $B_D(\mathbf{w})$ is not already large. Formally, we have the following result.

Theorem 7. *For any $\mathbf{w} \in \mathcal{W}$ such that $B_D(\mathbf{w}) \leq n$, we have $B_D^{\mathsf{DIM}}(\mathbf{w}) \geq B_D(\mathbf{w})$, where $\mathsf{DIM} \in \{\mathsf{drop}, \mathsf{sgld}, \mathsf{quant}\}$.*

5 DIFFERENTIAL GRADIENT DIVERSITY AND STABILITY

In this section, we turn to another important property of mini-batch SGD algorithm, *i.e.,* the generalization ability.

5.1 Stability and Generalization

Recall that in supervised learning, our goal is to learn a parametric model with small population risk $R(\mathbf{w}) := \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}[f(\mathbf{w};\mathbf{z})]$. In order to do so, we use empirical risk minimization, and hope to obtain a model that has both small empirical risk and small population risk to avoid overfitting. Formally, let A be a possibly randomized algorithm which maps the training data to the parameter space as $\mathbf{w} = A(\mathcal{S})$. In this paper, we use the model parameter obtained in the final iteration as the output of the mini-batch SGD algorithm, *i.e.,* $A(\mathcal{S}) = \mathbf{w}_T$. We define the *expected generalization error* of the algorithm as $\epsilon_{\text{gen}}(A) := \mathbb{E}_{\mathcal{S},A}[R_{\mathcal{S}}(A(\mathcal{S})) - R(A(\mathcal{S}))]$.

Bousquet and Elisseeff (2002) show the equivalence between the generalization error and algorithmic stability. The basic idea of proving generalization bounds using stability is to bound the distance between the model parameters obtained by running an algorithm on two datasets that only differ on one data point. This framework is used by Hardt et al. (2015) to show stability guarantees for serial SGD algorithm for Lipschitz and smooth loss functions. Roughly speaking, they show upper bounds $\overline{\gamma}$ on the step-size below which serial SGD is stable. This yields, as a corollary, that mini-batch SGD is stable provided the step-size is upper bounded by $\overline{\gamma}/B$. We remind the reader that since we absorb the $1/B$ factor in the step-size, the only step-size for which the analysis by Hardt et al. (2015) would imply stability for SGD is $1/B$ less than what we suggest in the convergence results. In the following parts of this section, we show that the mini-batch algorithm with a similar step-size to SGD is indeed stable, provided the *differential gradient diversity* is large enough.

[4]We note that our notion of DropConnect is slightly different from the original paper (Wan et al., 2013), but is of similar spirit.

5.2 Differential Gradient Diversity

The stability of mini-batch SGD is governed by the *differential gradient diversity*, defined as follows.

Definition 6 (differential gradient diversity and batch-size bound)**.** *For any $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$, $\mathbf{w} \neq \mathbf{w}'$, the differential gradient diversity and batch-size bound is given by*

$$\overline{\Delta}_D(\mathbf{w}, \mathbf{w}') := \frac{\sum_{i=1}^n \|\nabla f_i(\mathbf{w}) - \nabla f_i(\mathbf{w}')\|_2^2}{\|\sum_{i=1}^n \nabla f_i(\mathbf{w}) - \nabla f_i(\mathbf{w}')\|_2^2},$$
$$\overline{B}_D(\mathbf{w}, \mathbf{w}') := n\overline{\Delta}_D(\mathbf{w}, \mathbf{w}').$$

Although it is a distinct measure, differential gradient diversity shares similar properties with gradient diversity. For example, the lower bounds for $B_D(\mathbf{w})$ in examples 1 and 2 in Section 4.1 also hold for $\overline{B}_D(\mathbf{w}, \mathbf{w}')$, and two mechanisms, DropConnect and SGLD also induce differential gradient diversity, as we note in the supplementary material.

5.3 Stability of Mini-batch SGD

We analyze the stability (generalization) of mini-batch SGD via differential gradient diversity. We assume that, for each $\mathbf{z} \in \mathcal{Z}$, the loss function $f(\mathbf{w};\mathbf{z})$ is convex, L-Lipschitz and β-smooth in $\mathcal{W}$. We choose not to discuss the generalization error for nonconvex functions, since this may require a significantly small step-size (Hardt et al., 2015).

Our result is stated informally in Theorem 8, and holds for both convex and strongly convex functions. Here, $\overline{\gamma}$ is the step-size upper bound required to guarantee stability of serial SGD, and differently from the convergence results, we treat $\overline{B}_D(\mathbf{w}, \mathbf{w}')$ as a random variable defined by the sample $\mathcal{S}$.

Theorem 8 (informal stability result)**.** *Suppose that, with high probability, the batch-size $B \lesssim \overline{B}_D(\mathbf{w}, \mathbf{w}')$ for all $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$, $\mathbf{w} \neq \mathbf{w}'$. Then, after the same number of gradient updates, the generalization errors of mini-batch SGD and serial SGD satisfy $\epsilon_{gen}(\mathsf{minibatch\ SGD}) \lesssim \epsilon_{gen}(\mathsf{serial\ SGD})$, and such a guarantee holds for any step-size $\gamma \lesssim \overline{\gamma}$.*

Therefore, our main message is that, if with high probability, batch-size B is smaller than $\overline{B}_D(\mathbf{w}, \mathbf{w}')$ for all

$\mathbf{w}, \mathbf{w}'$, mini-batch SGD and serial SGD can be both stable in roughly the *same range* of step-sizes, and the generalization error of mini-batch SGD and serial SGD are roughly the *same*. We now provide the precise statements. In the following, we denote by $\mathbb{1}$ the indicator function.

Theorem 9 (generalization error of convex functions)**.** *Suppose that for any $\mathbf{z} \in \mathcal{Z}$, $f(\mathbf{w}; \mathbf{z})$ is convex, L-Lipschitz and β-smooth in $\mathcal{W}$. For a fixed step size $\gamma > 0$, let*

$$\eta = \mathbb{P}\Big\{ \inf_{\mathbf{w} \neq \mathbf{w}'} \overline{B}_D(\mathbf{w}, \mathbf{w}') < \frac{B-1}{\frac{2}{\gamma\beta} - 1 - \frac{1}{n-1}\mathbb{1}_{B>1}} \Big\},$$

where the probability is over the randomness of $\mathcal{S}$. Then the generalization error of mini-batch SGD satisfies $\epsilon_{gen} \leq 2\gamma L^2 \frac{T}{n}(1-\eta) + 2\gamma L^2 T\eta$.

It is shown by Hardt et al. (2015) that $\epsilon_{\text{gen}}(\text{serial SGD}) \leq 2\gamma L^2 \frac{T}{n}$, for convex functions, when $\gamma \leq \frac{2}{\beta}$. Notice that in our result, when $B = 1$, we get $\eta = 0$, and thus recover the generalization bound for serial SGD. Further, suppose one can find $\overline{B}$ such that $\inf_{\mathbf{w} \neq \mathbf{w}'} \overline{B}_D(\mathbf{w}, \mathbf{w}') \geq \overline{B}$ with high probability. Then by choosing $B \leq 1 + \delta\overline{B}$, and $\gamma \leq \frac{2}{\beta(1+\delta+\frac{1}{n-1})}$, we obtain similar generalization error as the serial algorithm without significant change in the step-size range. For strongly convex functions, we have:

Theorem 10 (generalization error of strongly convex functions)**.** *Suppose that for any $\mathbf{z} \in \mathcal{Z}$, $f(\mathbf{w}; \mathbf{z})$ is L-Lipschitz, β-smooth, and λ-strongly convex in $\mathcal{W}$, and $B \leq \frac{1}{2\gamma\lambda}$. For a fixed step size $\gamma > 0$, let*

$$\eta = \mathbb{P}\Big\{ \inf_{\mathbf{w} \neq \mathbf{w}'} \overline{B}_D(\mathbf{w}, \mathbf{w}') < \frac{B-1}{\frac{2}{\gamma(\beta+\lambda)} - 1 - \frac{1}{n-1}\mathbb{1}_{B>1}} \Big\},$$

where the probability is over the randomness of $\mathcal{S}$. Then the generalization error of mini-batch SGD satisfies $\epsilon_{gen} \leq \frac{4L^2}{\lambda n}(1-\eta) + 2\gamma L^2 T\eta$.

Again, as shown by Hardt et al. (2015), we have $\epsilon_{\text{gen}}(\text{serial SGD}) \leq \frac{4L^2}{\lambda n}$ for strongly convex functions, when $\gamma \leq \frac{2}{\beta+\lambda}$. Thus, our remarks for the convex case above also carry over here. We also mention that while in general, the probability parameter η may appear to weaken the bound, there are practical functions for which η has rate decaying in n. For example, for generalized linear functions, we can show that when the feature vectors have i.i.d. sub-Gaussian entries, choosing $B \lesssim d$ yields $\eta \lesssim ne^{-d}$, which has polynomial decay in n when $d = \Omega(\log(n))$. For details, see the supplementary material.

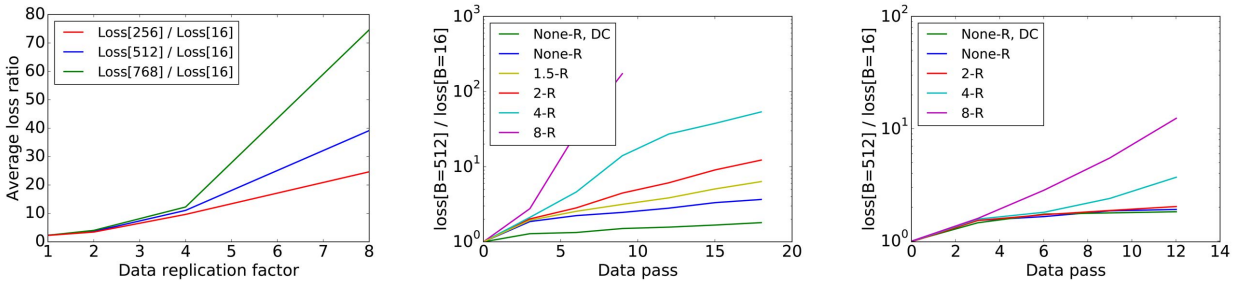
6 EXPERIMENTS

We conduct experiments to justify our theoretical results. Our neural network experiments are all implemented in Tensorflow and run on Amazon EC2 instances g2.2xlarge.

Convergence We conduct the experiments on a logistic regression model and two deep neural networks (a cuda convolutional neural network (Krizhevsky et al., 2012) and a deep residual network (He et al., 2016)) with cross-entropy loss running on CIFAR-10 dataset. These results are presented in Figure 2. We use data replication to implicitly construct datasets with different gradient diversity. By replication with a factor r (or r-replication), we mean picking a random $1/r$ fraction of the data and replicating it r times. Across all configurations of batch-sizes, we tune our (constant) step-size to maximize convergence, *e.g.,* to minimize training time. The sample size does not change by data replication, but gradient diversity conceivably gets smaller while we increase r. We use the ratio of the loss function for large batch-size SGD (*e.g.,* $B = 512$) to the loss for small batch-size SGD (*e.g.,* $B = 16$) to measure the negative effect of large batch sizes on the convergence rate. When this ratio gets larger, the algorithm with the large batch-size is converging slower. We can see from the figures that while we increase r, the large batch-size instances indeed perform worse, and the large batch instance performs the best when we have DropConnect, due to its diversity-inducing effect, as discussed in the previous sections. This experiment thus validates our theoretical findings.

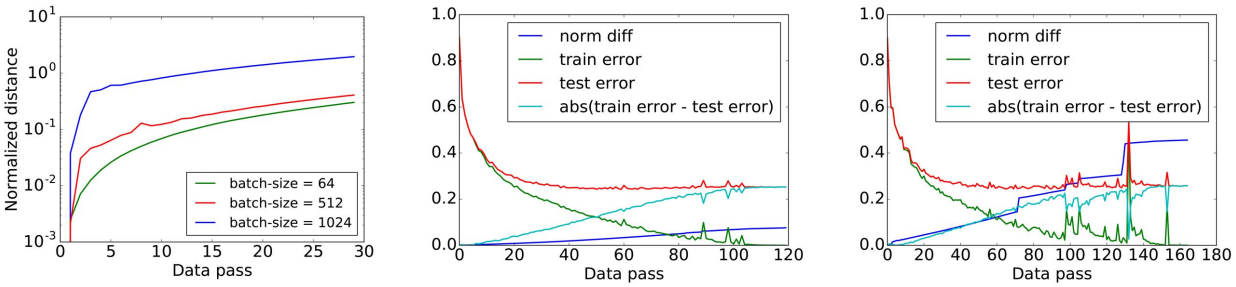
Stability We also conduct experiments to study the effect of large batch-size on the stability of mini-batch SGD. Our experiments essentially use the same technique as in the study for serial SGD by Hardt et al. (2015). Based on the CIFAR-10 dataset, we construct two training datasets which only differ in one data point, and train a cuda convolutional neural network using the same mini-batch SGD algorithm on these two datasets. For different batch-sizes, we test the normalized Euclidean distance $\sqrt{\|\mathbf{w} - \mathbf{w}'\|_2^2 / (\|\mathbf{w}\|_2^2 + \|\mathbf{w}'\|_2^2)}$ between the obtained model on the two datasets. As shown in Figure 3a, the normalized distance between the two models becomes larger when we increase the batch-size, which implies that we lose stability by having a larger batch-size. We also compare the generalization behavior of mini-batch SGD with $B = 512$ and $B = 1024$, as shown in Figures 3b and 3c. As we can see, for large batch sizes, the models exhibit higher variance in their generalization behavior, and our observation is in agreement with Keskar et al. (2016).

Diversity-inducing Mechanisms We finally implement diversity-inducing mechanisms in a distributed setting with 2 workers and test the speedup gains. We use a convolutional neural network on MNIST and implement the DropConnect mechanism with drop probability $p_{\text{drop}} = 0.4, 0.5$. We tune the step-size γ and batch-size B for vanilla mini-batch SGD and the diversity-induced setting, and find the



(a) (b) (c)

Figure 2: Data replication. Here, 2-R, 4-R, etc represent 2-replication, 4-replication, etc, and DC stand for DropConnect. (a) Logistic regression with two classes of CIFAR-10 (b) Cuda convolutional neural network (c) Residual network. For (a), we plot the average loss ratio during all the iterations of the algorithm, and average over 10 experiments; for (b), (c), we plot the loss ratio as a function of the number of passes over the entire dataset, and average over 3 experiments. We observe that with the larger replication factor, the gap of convergence increases.



(a) (b) (c)

Figure 3: Stability. (a) Normalized Euclidean distance vs number of data passes. (b) Generalization behavior of batch-size 512. (c) Generalization behavior of batch-size 1024. Results are averaged over 3 experiments.

(γ, B) pair that gives the fastest convergence for each setting. Then, we compare the overall run time to reach 90%, 95%, and 99% training accuracy. The results are shown in Table 1, where each time measurement is averaged over 5 runs. Comparing wall-clock times, we see DropConnect provides significant improvements. Indeed, the the batch-size gain afforded by DropConnect—the best batch-size for vanilla mini-batch SGD is 256, while with the diversity-inducing mechanism, it becomes 512—is able to dwarf the noise in gradient computation. Reducing communication cost thus has the biggest effect on runtime, more so than introducing additional variance in stochastic gradient computations.

Table 1: Speedup Gains via DropConnect

train accuracy (%)		90	95	99
mini-batch	time (sec)	46.97	57.39	361.52
$p_{\text{drop}} = 0.4$	time (sec)	24.88	39.12	313.60
	gain (%)	46.98	31.83	13.25
$p_{\text{drop}} = 0.5$	time (sec)	29.68	43.24	317.79
	gain (%)	36.76	24.66	12.09

7 CONCLUSION

We propose the notion of gradient diversity to measure the dissimilarity between concurrent gradient updates in mini-batch SGD. We show that, for both convex and nonconvex loss functions, the convergence rate of mini-batch SGD is identical—up to constant factors—to that of serial SGD, provided that the batch-size is at most proportional to a bound implied by gradient diversity. We also develop a corresponding lower bound for the convergence rate of strongly convex objectives. Our results show that on problems with high gradient diversity, the distributed implementation of mini-batch SGD is amenable to better speedups. We also establish similar results for generalization using the notion of differential gradient diversity. Some open problems include finding more mechanisms that improve gradient diversity, and in neural network learning, studying how the network structure, such as width, depth, and activation functions, impacts gradient diversity.

Acknowledgements

We gratefully acknowledge the support of the NSF through CIF award 1703678 and grant IIS-1619362, Berkeley DeepDrive Industry Consortium, Gift award from Huawei, and AWS Cloud Credits for Research.

References

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

D. Alistarh, J. Li, R. Tomioka, and M. Vojnovic. Qsgd: Randomized quantization for communication-optimal stochastic gradient descent. *arXiv preprint arXiv:1610.02132*, 2016.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2 (Mar):499–526, 2002.

J. Chen, R. Monga, S. Bengio, and R. Jozefowicz. Revisiting distributed synchronous sgd. *arXiv preprint arXiv:1604.00981*, 2016.

T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv:1512.01274*.

T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *11th USENIX OSDI 14*, pages 571–582, 2014.

A. Cotter, O. Shamir, N. Srebro, and K. Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *NIPS*, pages 1647–1655, 2011.

H. Daneshmand, A. Lucchi, and T. Hofmann. Starting small-learning with adaptive sample sizes. In *International conference on machine learning*, pages 1463–1471, 2016.

S. De, A. Yadav, D. Jacobs, and T. Goldstein. Big batch sgd: Automated inference using adaptive batch sizes. *arXiv preprint arXiv:1610.05792*, 2016.

J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al. Large scale distributed deep networks. In *NIPS*, pages 1223–1231, 2012.

O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13 (Jan):165–202, 2012.

A. Dfossez and F. Bach. Constant step size least-mean-square: Bias-variance trade-offs and optimal sampling distributions. *arXiv preprint arXiv:1412.0156*, 2014.

J. Duchi, M. I. Jordan, and B. McMahan. Estimation, optimization, and parallelism when data is sparse. In *NIPS*, pages 2832–2840, 2013.

M. P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3): A1380–A1405, 2012.

R. Frostig, R. Ge, S. M. Kakade, and A. Sidford. Competing with the empirical risk minimizer in a single pass. In *Conference on learning theory*, pages 728–763, 2015.

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

R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD*, pages 69–77. ACM, 2011.

S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

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

M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

E. Hoffer, I. Hubara, and D. Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *arXiv preprint arXiv:1705.08741*, 2017.

M. Jaggi, V. Smith, M. Takác, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan. Communication-efficient distributed dual coordinate ascent. In *NIPS*, pages 3068–3076, 2014.

P. Jain, S. M. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford. Parallelizing stochastic approximation through mini-batching and tail-averaging. *arXiv preprint arXiv:1610.03774*, 2016.

T. Joachims. Training linear svms in linear time. In *12th ACM SIGKDD*, pages 217–226. ACM, 2006.

H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Lojasiewicz condition. In

Joint Eur. Conf. on ML and Knowledge Disc. in Databases, pages 795–811. Springer, 2016.

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

J. Konečnỳ, J. Liu, P. Richtárik, and M. Takáč. Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):242–255, 2016.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

M. Li, T. Zhang, Y. Chen, and A. J. Smola. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD*, pages 661–670. ACM, 2014.

J. Liu, S. Wright, C. Re, V. Bittorf, and S. Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. In *Proceedings of ICML 14*, pages 469–477, 2014.

S. Lojasiewicz. A topological property of real analytic subsets. *Coll. du CNRS, Les Âťequations aux dÂťerivÂťees partielles*, pages 87âĂŞ–89, 1963.

D. Needell and R. Ward. Batched stochastic gradient descent with weighted sampling. *arXiv preprint arXiv:1608.07641*, 2016.

A. Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *NIPS*, pages 1574–1582, 2014.

F. Niu, B. Recht, C. Re, and S. Wright. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, pages 693–701, 2011.

B. T. Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963.

H. Qi, E. R. Sparks, and A. Talwalkar. Paleo: A performance model for deep neural networks. 2016.

S. Shalev-Shwartz and T. Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *NIPS*, pages 378–385, 2013.

M. Takác, A. S. Bijral, P. Richtárik, and N. Srebro. Mini-batch primal and dual methods for svms. In *ICML (3)*, pages 1022–1030, 2013.

M. Takáč, P. Richtárik, and N. Srebro. Distributed mini-batch sdca. *preprint arXiv:1507.08322*, 2015.

L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pages 1058–1066, 2013.

J. Wang, W. Wang, and N. Srebro. Memory and communication efficient distributed stochastic optimization with minibatch prox. *arXiv preprint arXiv:1702.06269*, 2017.

W. Wang and N. Srebro. Stochastic nonconvex optimization with large minibatches. *arXiv preprint arXiv:1709.08728*, 2017.

M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of ICML*, pages 681–688, 2011.

W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. *arXiv preprint arXiv:1705.07878*, 2017.

H. Yun, H.-F. Yu, C.-J. Hsieh, S. Vishwanathan, and I. Dhillon. Nomad: Non-locking, stochastic multi-machine algorithm for asynchronous and decentralized matrix completion. *arXiv:1312.0193*, 2013.

C. Zhang, H. Kjellstrom, and S. Mandt. Stochastic learning on imbalanced data: Determinantal point processes for mini-batch diversification. *arXiv preprint arXiv:1705.00607*, 2017.

P. Zhao and T. Zhang. Accelerating minibatch stochastic gradient descent using stratified sampling. *arXiv preprint arXiv:1405.3080*, 2014.