BALANCING RATES AND VARIANCE VIA ADAPTIVE BATCH-SIZES IN FIRST-ORDER STOCHASTIC OPTIMIZATION

Zhan Gao[†]*, Alec Koppel*[‡]*, and Alejandro Ribeiro*[†]

[†]Department of Electrical and Systems Engineering, University of Pennsylvania, USA
[‡]Computational and Information Sciences Directorate, U.S. Army Research Laboratory, USA
Email: {gaozhan, akoppel, aribeiro}@seas.upenn.edu

ABSTRACT

Stochastic gradient descent is a canonical tool for addressing stochastic optimization problems, and forms the bedrock of modern machine learning and statistics. In this work, we seek to balance the fact that attenuating step-sizes is required for exact asymptotic convergence with the fact that larger constant step-sizes learn faster in finite time up to an error. To do so, rather than fixing the mini-batch and step-size at the outset, we propose a strategy to allow parameters to evolve adaptively. Specifically, the batch-size is set to be a piecewise-constant increasing sequence where the increase occurs when a suitable error criterion is satisfied. Moreover, the step-size is selected as that which yields the fastest convergence. The overall algorithm, two scale adaptive (TSA) scheme, is shown to inherit the exact asymptotic convergence of stochastic gradient method. More importantly, the optimal error decreasing rate is achieved theoretically, as well as an overall reduction in sample computational cost. Experimentally, we observe a favorable tradeoff relative to standard SGD schemes absorbing their advantages, which illustrates the significant performance of proposed TSA scheme.

Index Terms— Stochastic optimization, stochastic gradient descent, adaptive batch-size, optimal step-size

1. INTRODUCTION

Many machine learning [2, 3], control [4], and signal processing tasks [5] may be formulated as finding the minimizer of an expected cost parameterized by a random variable that encodes data. In particular, communications channel estimation [6], learning model mismatch of a dynamical system [7], and training modern vision systems [8], hinge upon solving stochastic optimization problems. Stochastic gradient descent (SGD) is a widely used approach for this problem [9, 10], but its practical parameter tuning can be difficult. This is because attenuating step-sizes is required for exact solutions, which slows learning to a stand still. Moreover, mini-batch sizes are typically fixed at the outset of training, which reduces variance by a fixed amount. In this work we balance the choice of step and mini-batch sizes to learn at fast rates with ever-decreasing variance as a consequence of allowing batch sizes to enlarge.

To contextualize our approach, we begin by noting that gradient methods yield the exact solutions for convex minimization problems [11, 12]. However, when the objective takes the form of an expected value, evaluating the gradient direction is intractable. To surmount this issue, SGD descends along stochastic gradients in lieu of true gradients, which are simply gradients evaluated at a single (or possibly multiple) sample point(s) [13–16]. Its simplicity and strong theoretical guarantees make it an attractive option, but its practical usage requires parameter tuning that can be unintuitive.

Specifically, it is well-understood that attenuating step-sizes is theoretically required for exact convergence, at the cost of reducing the learning speed to null as time progresses [17, 18]. Experimentally, constant step-sizes vastly improve performance, but only converge approximately [19]. The choices of step-size in terms of learning rate typically depend on the Lipschitz modulus of continuity and strong convexity parameter, which then determines the asymptotic mean square error. Hypothetically, one would like to preserve a fast rate while ever-tightening the radius of convergence, such that exact convergence is obtained in the limit.

To do so, we shift focus to mini-batching. Mini-batching is a procedure where the stochastic gradient is averaged over multiple samples per iteration. Under constant learning rates, its practical effect is to reduce the variance of stochastic approximation error, which tightens asymptotic convergence [20–22]. Intriguingly, it has recently been established that when the batch-size grows geometrically with the iteration index, SGD obtains exact solutions in the limit even under fixed step-sizes [23]. This fact then motivates allowing the batch-size to grow as slowly as possible, while maintaining an optimal constant learning rate determined by problem smoothness. Doing so is exactly the proposed parameter selection strategy of this work, which we establish yields an overall reduction in sample computational complexity relative to standard approaches.

In particular, in Section 2, we clarify the problem definition, and give preliminary analysis for SGD. In Section 3 we propose our optimal parameter selection algorithm, two scale adaptive (TSA) scheme, that outperforms standard SGD approaches for stochastic optimization problems, by taking both batch-size and step-size into consideration simultaneously. In Section 4, we establish our main result, which is an overall reduction in sample complexity of TSA scheme under optimal learning rates, equipped with its corresponding convergence analysis. We observe experimentally fast convergence rates with attenuating variance in Section 5 on a classification problem of hand-written digits. In particular, by choosing parameters in this way, we minimize both asymptotic bias and sample path variance of learning. Lastly, we conclude in Section 6.

2. PROBLEM FORMULATION

Denote as $\mathbf{x} \in \mathbb{R}^p$ a decision vector to be determined, e.g., the parameters of a statistical model, $\boldsymbol{\xi} \in \Omega$ a random variable with the probability distribution $\mathbf{p}$, and $f(\mathbf{x}, \boldsymbol{\xi})$ an objective function whose average minimizer we seek to compute. We consider the stochastic

Proofs may be found in [1].

optimization problem defined as

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} \int_{\Omega} f(\mathbf{x}, \boldsymbol{\xi}) p(d\boldsymbol{\xi}) = \min_{\mathbf{x}} \mathbb{E}[f(\mathbf{x}, \boldsymbol{\xi})]. \quad (1)$$

Typically, the distribution $\mathbf{p}$ of $\boldsymbol{\xi}$ is unknown, and therefore the expectation $F(\mathbf{x})$ in (1) is not computable. This issue usually motivates drawing N samples $\{\boldsymbol{\xi}_i\}_{i=1}^N$ from $\mathbf{p}$ and solving the corresponding empirical risk minimization (ERM) problem [24]

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}, \boldsymbol{\xi}_i) \ . \quad (2)$$

We set our focus on the population problem (1) in this paper. Define the simplifying notation $f_i(\mathbf{x}) := f(\mathbf{x}, \boldsymbol{\xi}_i)$ and $\nabla f_S(\mathbf{x})$ as the average gradient of a sample set $\mathcal{S} = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n\}$, i.e., $\nabla f_S(\mathbf{x}) = \frac{1}{n} \sum_{i \in S} \nabla f_i(\mathbf{x})$, which is commonly referred to as the mini-batch gradient. Subsequently, we require the following assumptions.

Assumption 1. *The gradient of expected objective function $\nabla F(\mathbf{x})$ is Lipschitz continuous, i.e., for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$,*

$$\| \nabla F(\mathbf{x}) - \nabla F(\mathbf{y}) \|_2 \leq L \| \mathbf{x} - \mathbf{y} \|_2, \quad (3)$$

where $\| \cdot \|_2$ is the (2-) norm and L is the Lipschitz constant.

Assumption 2. *All objective functions $\{f_i(\mathbf{x})\}$ are differentiable, and the expected objective function $F(\mathbf{x})$ is ℓ-strongly convex.*

Assumption 3. *There exists a constant w such that for any $\mathbf{x} \in \mathbb{R}^p$, the stochastic objective function $f_i(\mathbf{x})$ has*

$$\| Var[\nabla f_i(\mathbf{x})] \|_1 \leq w, \quad (4)$$

where $\| \cdot \|_1$ is the (1-) norm and $Var[\nabla f_i(\mathbf{x})]$ is the variance vector, each component of which is the variance of corresponding component of vector $\nabla f_i(\mathbf{x})$.

Note that Assumptions 1-3 are mild and common in optimization analysis and can be satisfied in various practical problems. Additionally, under these conditions, the minimizer $\mathbf{x}^*$ of (1) is unique.

Now consider the standard mini-batch stochastic gradient descent (SGD) algorithm given by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f_{S_k}(\mathbf{x}_k). \quad (5)$$

Here, α_k is the step-size and S_k is the mini-batch at k-th iteration. We focus on optimal selections of batch-size $n_k := |S_k|$ and step-size α_k to reduce the computation but preserve the convergence.

3. ALGORITHM

We begin by establishing a proposition that characterizes the convergence rate of (5) in expectation related to the batch-size n_k and step-size α_k. Doing so forms the foundation upon which we develop our optimal parameter tuning algorithm.

Proposition 1. *Consider the SGD given by* (5)*, under Assumptions 1-3, with constant step-size $\alpha_k = \alpha$ and batch-size $n_k = n$. The sequence of expected objective function errors has*

$$\begin{aligned} &\mathbb{E}\left[F(\mathbf{x}_{k+1}) - F(\mathbf{x}^*)\right] \\ &\leq \underbrace{r(\alpha)^{k+1} \left(F(\mathbf{x}_0) - F(\mathbf{x}^*)\right)}_{:=Q_1} + \underbrace{\frac{\alpha L w}{2n(2\ell - L\ell\alpha)}}_{:=Q_2}, \end{aligned} \quad (6)$$

where $r(\alpha) = 1 - 2\ell\alpha + L\ell\alpha^2$ is the approximate modulus of contraction and the expectation $\mathbb{E}[\cdot]$ is over randomness in S_k.

Proposition 1 establishes the dependence of convergence on problem constants, step-size and batch-size selections. In particular, the bound in (6) consists of two terms. Q_1, the convergence rate term, decreases with iteration k provided that $r(\alpha) < 1$. The error neighborhood term Q_2 typically determines the limiting radius of convergence for constant step-size, and is associated with the variance of stochastic approximation (sub-sampling) error. We observe an inverse dependence on the batch-size n with respect to Q_2. Our algorithm, the two scale adaptive (TSA) scheme, is then proposed by exploiting the structure of Q_1 and Q_2 to reduce the overall computational complexity of (5) required to solve (1).

3.1. Two Scale Adaptive Scheme

Note from (6), Q_1 decreases monotonically while Q_2 stays constant with increasing iterations. Once Q_1 decays to be smaller than Q_2, $\mathbf{x}_k$ cannot converge to a tighter neighborhood than that determined by Q_2. Therefore, in order to continue tightening the radius of convergence, one must reduce Q_2 by either decreasing the step-size α or increasing the batch-size n. The TSA scheme gives a strategy about when and how to make this change. The algorithm is divided into two stages: the inner-scale stage performs SGD with constant step-size α and batch-size n, and the outer-scale stage tunes α, n to tighten the radius of convergence, which are formally introduced below.

Initialization: With initial step-size $\alpha = \alpha_0$ and batch-size $n = n_0$, the corresponding Q_1^0 and Q_2^0 are defined as (6). Without loss of generality, assume $\mathbb{E}\left[F(\mathbf{x}_0) - F(\mathbf{x}^*)\right]$ is large such that $Q_1^0 \geq Q_2^0$ initially, and then Q_1^0 dominates. Note that the decreasing rate of Q_1^0 is $r(\alpha_0)$, which is a quadratic function of the step-size α_0. Thus, to ensure an optimal decrease, $\alpha_0 = 1/L$ is selected to achieve its minimal value $1 - \ell/L$. The corresponding Q_2^0 is given by

$$Q_2^0 = \frac{\frac{1}{L} L w}{2n_0(2\ell - L\ell\frac{1}{L})} = \frac{w}{2n_0\ell}. \quad (7)$$

Furthermore, convergence rate is an essential attribute to quantify algorithm performance, which henceforth motivates fixing the optimal step-size $\alpha = 1/L$. Doing so ensures the fastest decrease of Q_1 through all iterations, during which we propose evolving the batch-size n to tighten the error neighborhood Q_2. Proceeding from this initialization, we then discuss how to progressively enlarge the mini-batch size.

(1) *Inner-scale stage*. At t-th inner-scale stage, let $\alpha_t = 1/L$ and n_t be the current step-size and batch-size, and K be the beginning number of iteration. Follow the SGD with constant α_t and n_t, and define Q_1^t and Q_2^t as

$$Q_1^t = \left(1 - \frac{\ell}{L}\right)^{k_t} \mathbb{E}\left[F(\mathbf{x}_K) - F(\mathbf{x}^*)\right], \quad (8)$$

$$Q_2^t = \frac{\alpha_t L w}{2n_t(2\ell - L\ell\alpha_t)} = \frac{w}{2n_t\ell}, \quad (9)$$

where $k_t > 0$ is the passed number of iterations at t-th inner-scale stage. As k_t increases, Q_1^t decreases multiplied by $1 - \ell/L$ recursively, while Q_2^t stays constant. Then there exists K_t such that

$$K_t = \max_{k_t} \left\{Q_1^t \geq Q_2^t\right\}. \quad (10)$$

K_t is the largest iteration before Q_1^t drops below Q_2^t and is named as the duration of t-th inner-scale stage.

Due to the existence of $\mathbb{E}\left[F(\mathbf{x}_K) - F(\mathbf{x}^*)\right]$ where $\mathbf{x}_K$ is unknown, (10) cannot be directly used as a criterion indicating the end

of inner-scale stage. Let $\{\alpha_0, \alpha_1, ..., \alpha_{t-1}\}$, $\{n_0, n_1, ..., n_{t-1}\}$ and $\{K_0, K_1, ..., K_{t-1}\}$ be step-sizes, batch-sizes and durations of previous $t-1$ inner-scale stages, and we have $\alpha_0 = \cdots = \alpha_{t-1} = 1/L$ and $K = \sum_{i=0}^{t-1} K_i$. Utilizing (8) then yields:

$$\begin{aligned}
&\mathbb{E}\left[(F(\mathbf{x}_K) - F(\mathbf{x}^*))\right] = \mathbb{E}\left[\left(F(\mathbf{x}_{\sum_{i=0}^{t-1} K_i}) - F(\mathbf{x}^*)\right)\right] \\
&\leq \left(1 - \frac{\ell}{L}\right)^{K_{t-1}} \mathbb{E}\left[F(\mathbf{x}_{\sum_{i=0}^{t-2} K_i}) - F(\mathbf{x}^*)\right] + Q_2^{t-1} \\
&\leq 2\left(1 - \frac{\ell}{L}\right)^{K_{t-1}} \mathbb{E}\left[F(\mathbf{x}_{\sum_{i=0}^{t-2} K_i}) - F(\mathbf{x}^*)\right].
\end{aligned} \tag{11}$$

The last inequality follows from the definition of K_{t-1} (10). Recursively applying this property, we can obtain

$$Q_1^t \leq 2^t \left(1 - \frac{\ell}{L}\right)^{\sum_{i=0}^{t-1} K_i + k_t} (F(\mathbf{x}_0) - F(\mathbf{x}^*)). \tag{12}$$

The criterion (10) can then be alternated as

$$K_t = \max_{k_t} \left\{ 2^t \left(1 - \frac{\ell}{L}\right)^{\sum_{i=0}^{t-1} K_i + k_t} (F(\mathbf{x}_0) - F(\mathbf{x}^*)) \geq \frac{w}{2n_t \ell} \right\}. \tag{13}$$

Overall, we run SGD with step-size $\alpha_t = 1/L$ and batch-size n_t for K_t iterations to reduce Q_1^t. Once the total iteration count reaches $\sum_{i=0}^{t} K_i$, then we augment the batch-size, as we detail in the t-th outer-scale stage of algorithm.

(2) *Outer-scale stage*. At t-th outer-scale stage, we evolve parameters to reduce the error neighborhood term Q_2^t, which can be realized either by the decreasing of α_t or the increasing of n_t. However, the former slows down the convergence rate $r(\alpha_t)$, while the latter increases the sample computational complexity. The tradeoff between these two factors needs to be judiciously balanced.

As mentioned in Initialization, the step-size is fixed as $\alpha_t = \alpha_0 = 1/L$ to maintain the optimal decrease of Q_1 in our algorithm. Then, we increase the current batch-size n_t to n_{t+1} in one of two possible ways: additivity or multiplicativity.

$$n_{t+1} = n_t + \beta_t, \quad \beta_t \geq 1, \tag{14}$$

$$n_{t+1} = m_t n_t, \quad m_t > 1, \tag{15}$$

where β_t and m_t are additive and multiplicative parameters. Though the selection of (14) and (15) is not the key point in the TSA scheme, it is still an available tradeoff which can be tuned in practice to help improve performance. So far t-th outer-scale stage has been completed, and $(t+1)$-th inner-scale stage follows recursively.

Here β_t or m_t is selected appropriately to ensure each K_t defined in (13) larger than 0. Together, the TSA scheme is summarized in Algorithm 1 where k is the number of total iterations and t is the number of inner/outer-scale stages. Note that in practice, by assuming $|F(\mathbf{x})|$ is bounded, $F(\mathbf{x}_0) - F(\mathbf{x}^*)$ can be approximated by $\max_{\mathbf{x},\mathbf{y}} |F(\mathbf{x}) - F(\mathbf{y})|$ initially.

4. PERFORMANCE ANALYSIS

In this section, we present performance analysis for the proposed TSA scheme with respect to its convergence and sample complexity. In particular, we establish that it inherits the limiting convergent properties of SGD while reducing the number of training samples required to reach an ϵ-suboptimal solution.

Algorithm 1 Two Scale Adaptive Scheme

Input: objective functions $\{f(\mathbf{x}, \boldsymbol{\xi})\}$, decision vector $\mathbf{x}_0$
Set step-size $\alpha = 1/L$; sample-size $|S_0| = n_0$ and $t = 0$
Compute $Q_1^t = F(\mathbf{x}_0) - F(\mathbf{x}^*)$, $Q_2^t = \frac{w}{2\ell n_0}$
for $k = 1, 2...$ **do**
  Update the decision vector $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f_{S_k}(\mathbf{x}_k)$
  Compute $Q_1^t = \left(1 - \frac{\ell}{L}\right) Q_1^t$
  if $\left(1 - \frac{\ell}{L}\right) Q_1^t \leq Q_2^t$ **then**
    Update batch-size $n_{t+1} = n_t + r_t$ or $n_{t+1} = m_t n_t$
    Update $Q_1^{t+1} = 2Q_1^t$, $Q_2^{t+1} = \frac{n_t Q_2^t}{n_{t+1}}$, $t = t + 1$
  end if
end for

4.1. Convergence Analysis

We first show that the sequence of expected objective values $F(\mathbf{x}_k)$ generated by the TSA scheme approaches the optimal value $F(\mathbf{x}^*)$ with the following Theorem 1. It guarantees the exact convergence of TSA, which validates the usefulness of algorithm.

Theorem 1. *Consider the TSA scheme. If the objective functions satisfy Assumptions 1-3, both sequences of $F(\mathbf{x}_k)$ and $\mathbf{x}_k$ converge to the optimal $F(\mathbf{x}^*)$ and $\mathbf{x}^*$ almost surely, i.e.,*

$$\begin{aligned}
&\lim_{k\to\infty} \mathbb{E}\left[F(\mathbf{x}_k) - F(\mathbf{x}^*)\right] = 0, \\
&\lim_{k\to\infty} \mathbb{E}\left[\| \mathbf{x}_k - \mathbf{x}^* \|_2\right] = 0.
\end{aligned} \tag{16}$$

Theorem 1 establishes the fact that the TSA scheme inherits the asymptotic convergence behavior of SGD with attenuating step-size selection. However, we note that this result is somewhat surprising since TSA attains exact convergence with a constant step-size, whereas SGD converges to an error neighborhood under this setting. This result is a precursor to the characterization of the overall sample complexity of Algorithm 1, which we do in the following.

4.2. Sample Complexity Reduction

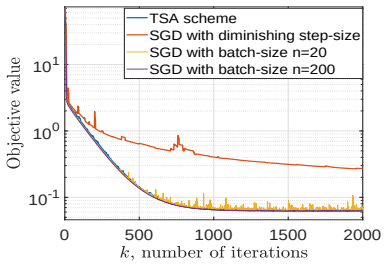
A critical benefit of TSA is its sample complexity reduction. For the convenience of comparison, we hypothesize that SGD uses the same optimal step-size $\alpha = 1/L$ as TSA. Moreover, TSA is with the multiplicative rule for augmenting the batch-size (15) and $m_t = m$ for all outer-scale stages. Under these conditions, the following sample complexity characterization of TSA compared with SGD holds.

Theorem 2. *Consider the TSA scheme starting with the initial batch-size $n_0 = 1$, and the SGD with constant step-size $\alpha = 1/L$ and batch-size n. Assume that the objective functions satisfy Assumptions 1-3 and define the initial error as $D := F(\mathbf{x}_0) - F(\mathbf{x}^*)$. To achieve an ϵ-suboptimal solution, the ratio γ between the number of training samples required for TSA and SGD is bounded as*

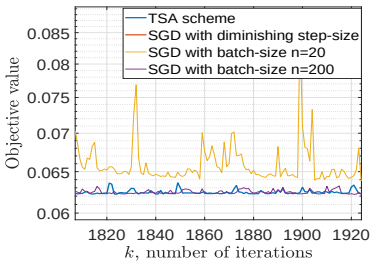
$$\gamma \leq \frac{m \left\lceil \log_{1-\frac{\ell}{L}} \frac{L-\ell}{2mL} \right\rceil}{(m-1) \left\lceil \log_{1-\frac{\ell}{L}} \frac{\epsilon}{2D} \right\rceil} + \mathcal{O}(\epsilon) \tag{17}$$

where $\lceil \cdot \rceil$ is the ceil function.

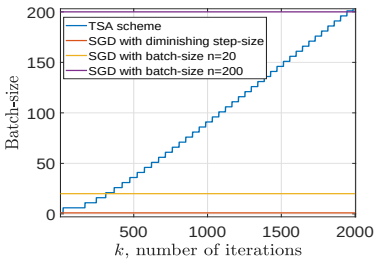
Observe from Theorem 2, the ratio γ of sample complexity of TSA to SGD, also known as the relative efficiency, is approximately proportional to $\mathcal{O}(-1/\log \epsilon) + \mathcal{O}(\epsilon)$, meaning that for accurate solutions, i.e., ϵ close to null, both terms approach 0, and a significant



(a) Objective vs. iteration (overall figure)



(b) Objective vs. iteration (larger figure)



(c) Batch-size vs. iteration

Fig. 1: MNIST logistic regression: TSA scheme, SGD with constant batch-size $n = 200$, and SGD with constant batch-size $n = 20$.

reduction in the number of required samples is attained. We provide a detailed analysis for the special case when $m = 2$. For a small ϵ, the second term $\mathcal{O}(\epsilon)$ is negligible. Moreover, the simplified expression of first term indicates that provided the rate $1 - \ell/L < 1$, the ratio γ is less than 1 when $\epsilon/(2D) \leq (1 - \ell/L)^2/16$, i.e., $\gamma \leq 1$ if $\epsilon \leq D(1 - \ell/L)^2/8$, which is almost always true unless the initial point is very close to the optimizer []. Therefore, in practice, the sample complexity of TSA is largely reduced compared with SGD, while both of them achieve the same suboptimal solution. The magnitude of the reduction is subtle and depends on problem constants, but at least we obtain a saving proportional to the sum of minus inverse of logarithmic factor of ϵ and ϵ, which may be substantial.

Overall, the TSA scheme provides a strategy to select the batch-size and step-size to preserve the optimal convergence rate while repeatedly reducing the stochastic approximation variance during learning. The batch-size increases only when necessary, allowing the reduced sample complexity relative to classical SGDs.

5. NUMERICAL EXPERIMENTS

In this section, we present experiments to demonstrate the performance of the TSA scheme compared with standard SGD schemes. Note that without particular descriptions, the default step-size of SGD is the same optimal step-size as TSA for clear comparison.

The visual classification problem of hand-written digits is considered, and here we focus on classifying digits 0 and 8. Given the training set $\mathcal{T} = \{(\mathbf{y}_n, z_n)\}_{n=1}^N$, let $\mathbf{y}_n \in \mathbb{R}^p$ be the feature vector of digit image and $z \in \{-1, 1\}$ be the label indicating which class (0 or 8) it belongs to. We formulate the problem as a logistic regression whose purpose is to train a hand-written digit classifier $\mathbf{x} \in \mathbb{R}^p$ to model the map between $\mathbf{y}$ and z. The instantaneous objective function $f(\mathbf{x}, \boldsymbol{\xi}) = f(\mathbf{x}, \mathbf{y}, z)$ in (1) is defined as

$$f(\mathbf{x}, \boldsymbol{\xi}) = \frac{\lambda}{2} \parallel \mathbf{x} \parallel_2^2 + \frac{1}{N} \sum_{n=1}^{N} \log\left(1 + \exp(-z_n \mathbf{x}^\top \mathbf{y}_n)\right), \quad (18)$$

where $(\lambda/2) \parallel \mathbf{x} \parallel_2^2$ is the regularization term. Note that it is actually an ERM, the approximation of stochastic optimization problem. We simulate the TSA scheme in an ERM for convenience, but it indeed is designed for all general stochastic optimization problems.

MNIST dataset is used where the dimension of features is $p = 784$ and total sample number is $N = 26491$. In this case, the variance of stochastic approximation is small such that we do not need a large batch-size to approximate the true gradient. Hence, the additive way (14) is chosen to increase the batch-size in TSA, and let $\beta_t = 5$. Fig. 1 compares the TSA scheme with three standard algorithms: SGD with diminishing step-size, SGDs with constant batch-sizes $n = 20$ and $n = 200$. Fig. 1 (a) depicts the objective value versus iterations of these four algorithms, and (b) is the larger figure illustrating performance differences more clearly. TSA and SGD with diminishing step-size both enjoy the exact convergence, while TSA outperforms with its fast decreasing rate by keeping the optimal step-size. As for comparisons with two approximate convergence algorithms, TSA and SGD with $n = 200$ show almost same good performances for the objective value convergence. However, SGD with $n = 20$ ranges rise and fall in a large error neighborhood, and performs worse than another two algorithms. Fig. 1 (c) displays the batch-size required at each iteration for them. Observe that TSA saves more than half of sample computational cost compared with SGD with $n = 200$, but shows a similar performance. As for SGDs with diminishing step-size and $n = 20$, though they show less sample complexity, they perform badly. Overall, TSA well balances another three algorithms and absorbs their advantages, such that converges fast and exactly, but with relatively low sample complexity.

Table 1: Number of training samples required to reduce the loss below 0.0622 for three algorithms: TSA, SGD with $n_k = 200$ and SGD with $n = 20$.

	Number of required samples
TSA	55651
SGD with $n = 200$	111500
SGD with $n = 20$	∞

From another perspective, we compare the number of samples required to reduce the objective (loss) below a certain desired value for above three algorithms, except SGD with diminishing step-size that converges too slowly to consider. Let the target loss value be 0.0622, and Table I shows the number of total samples required for them to drop the loss below this value. Note that for SGD with $n = 20$, it can never obtain a loss lower than 0.0622 due to its large approximation variance such that the number of samples is infinity.

6. CONCLUSIONS

This paper investigates the stochastic optimization problem that is of critical importance in wide science and engineering areas. The two scale adaptive (TSA) scheme is developed by co-considering batch-size and step-size of SGD simultaneously. Specifically, the optimal step-size is selected to acquire theoretically largest learning rate, while the batch-size is increased adaptively according to the proposed criterion. Equipped with the exact convergence, TSA has the fast rate due to the selected optimal step-size. At the same time, it only increases the batch-size when necessary, which reduces the sample complexity as much as possible. Numerical simulations are performed to show significant performance of TSA, which achieves a good balance among standard SGD schemes.

7. REFERENCES

[1] Z. Gao, A. Koppel, and A. Ribeiro, *U.S. Army Research Laboratory/University of Pennsylvania Technical Report, 2020.*, https://koppel.netlify.com/assets/papers/2020_report_zhan_etal.pdf.

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2nd edition, 2009.

[3] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[4] H. Pham, *Continuous-time Stochastic Control and Optimization with Financial Applications*. Springer, 2009.

[5] M. A. Pereyra, P. Schniter, E. Chouzenoux, J. Pesquet, J. Tourneret, A. O. Hero, and S. Mclaughlin, "A survey of stochastic simulation and optimization methods in signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 224–241, Nov. 2015.

[6] A. Ribeiro, "Ergodic stochastic optimization algorithms for wireless communication and networking," *IEEE Transactions on Signal Processing*, vol. 58, no. 12, pp. 6369–6386, 2010.

[7] A. Koppel, B. M. Sadler, and A. Ribeiro, "Proximity without consensus in online multi-agent optimization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[8] S. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[9] A. Mokhtari, A. Koppel, and A. Ribeiro, "Doubly random parallel stochastic methods for large scale learning," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 4847–4852.

[10] A. Mokhtari, A. Koppel, G. Scutari, and A. Ribeiro, "Large-scale nonconvex stochastic optimization by doubly stochastic successive convex approximation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4701–4705.

[11] L. A. Cauchy, "Methodes generales pour la resolution des systems d equations simultanees," *Comptes Rendus.de l Academie.des Sciences*, vol. 25, no. 2, pp. 536–538, 1847.

[12] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer, 2004.

[13] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[14] A. Ruszczynski and W. Syski, "Stochastic approximation method with gradient averaging for unconstrained problems," *IEEE Transactions on Automatic Control*, vol. 28, no. 12, pp. 1097–1105, 1983.

[15] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *International Conference on Neural Information Processing Systems*, 2004.

[16] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *International Conference on Computational Statistics*, 2010.

[17] Y. Wardi, "A stochastic algorithm using one sample point per iteration and diminishing stepsizes," *Journal of Optimization Theory and Applications*, vol. 61, no. 3, pp. 473–485, 1989.

[18] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[19] A. Nedic and D. Bertsekas, "Convergence rate of incremental subgradient algorithms," *Stochastic Optimization: Algorithms and Applications*, pp. 223–264, 2001.

[20] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *International Conference on Knowledge Discovery and Data Mining*, 2014.

[21] A. Cotter, O. Shamir, N. Srebro, and K. Sridharan, "Better mini-batch algorithms via accelerated gradient methods," in *International Conference on Neural Information Processing Systems*, 2011.

[22] J. Konecny, J. Liu, P. Richtarik, and M. Takac, "Mini-batch semi-stochastic gradient descent in the proximal setting," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 242–255, 2015.

[23] R. H. Byrd, G. M. Chin, J. Nocedal, and W. Y., "Sample size selection in optimization methods for machine learning," *Mathematical Programming*, vol. 134, no. 1, pp. 127–155, 2012.

[24] G. King and L. Zeng, "Logistic regression in rare events data," *Political Analysis*, vol. 9, no. 2, pp. 137–163, 2000.