Expert Systems With Applications 112 (2018) 1–14

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

To regularize or not: Revisiting SGD with simple algorithms and experimental studies

Wenwu He[a,b,c,*], Yang Liu[d]

[a] School of Mathematics and Physics, Fujian University of Technology, No. 3 Xueyuan Road, University Town, Minhou, Fuzhou, Fujian 350118, China
[b] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, No. 3 Xueyuan Road, University Town, Minhou, Fuzhou, Fujian 350118, China
[c] Fujian Collaborative Innovation Center for Beidou Navigation and Intelligent Traffic, No. 3 Xueyuan Road, University Town, Minhou, Fuzhou, Fujian 350118, China
[d] School of Engineering and Applied Science, Harvard University, 33 Oxford Street, MD 110, Cambridge, MA 02138, United States of America

ARTICLE INFO

Article history:
Received 16 August 2017
Revised 25 May 2018
Accepted 10 June 2018
Available online 15 June 2018

Keywords:
Stochastic gradient descent
Regularization
Accumulated stochastic gradient
Big data
SVMs
Regularized regression

ABSTRACT

Stochastic Gradient Descent (SGD) is one of the most popular first-order methods to solve optimization problems in large-scale, which has also been widely studied in intelligence system, deep neural networks, or machine learning. In this paper we set out to revisit SGD with practical concerns in mind and hope to provide intuition on how SGD should be done in the right way in applications for expert and intelligence systems.

In literature, when implementing SGD, regularization is often added to the loss function, to avoid ill-conditioning or overfitting, or to obtain solutions with desirable sparse structure. Intuitively, regularized loss function deviates from the true loss and in this case SGD may lead to a suboptimal solution. In this paper we revisit the matter of l_2 regularization for SGD, to investigate whether or when an explicit regularizer is necessary to obtain the desirable performance. We further introduce a simple stochastic algorithm (ASG) using the accumulated stochastic gradient of the un-regularized loss. Then experiments are carried out on benchmark data sets to validate the theoretical analysis.

The findings show that for l_2 regularization, (1) SGD without explicit regularizing (SGDE) actually possesses an implicit regularizer, and in the sense of upper bound of the convergence rate, it outperforms SGD regularized explicitly (SGDER) with a constant advantage; (2) ASG without explicit regularizing outperforms both SGDE and SGDER, especially in the case where the number of iteration T cannot be pre-specified; (3) for SGD algorithms, the schemes that places flexible weight on the output of the latest iteration can give a better trade-off, compared with the scheme using the output of the last iteration or taking the average of the output of each iteration, and this suggests that in application a tunable averaging scheme is preferable.

This study provides insights on using SGD algorithms for big data applications, e.g., to accelerate SVMs or regularized regression in large-scale, or to improve the performance of online learning or real time forecasting (control). In particular, when T is pre-specified, SGDE (without an explicit regularizer) can give us a well enough perforce with simplicity and understandability; when T cannot be pre-specified, ASG can improve the performance of standard SGD algorithms.

1. Introduction

To ability to learn from observations, to classify and to make decisions accordingly is crucial for many expert systems, such as to predict quality of a manufactory system, to achieve real-time monitoring of a control system, to predict and assess the health conditions of a patient, among many others. This ability becomes more challenging to achieve when we consider a large-scale learning problem, many often due to the requirement of heavy computations. In this paper, we revisit the classical stochastic gradient descent algorithms (SGD) method, which is widely deployed for the purpose of optimization and classifier training in these systems.

As a general optimizing solver, stochastic gradient descent (SGD) (Borkar, 2008; Léon, 2012) and its varieties have received great interests in a wide range of literature, including informed

* Corresponding author at: School of Mathematics and Physics, Fujian University of Technology, No. 3 Xueyuan Road, University Town, Minhou, Fuzhou, Fujian 350118, China.
E-mail addresses: hwwhbb@163.com (W. He), yangl@seas.harvard.edu (Y. Liu).

https://doi.org/10.1016/j.eswa.2018.06.026
0957-4174/

search (Russell & Norvig, 2016; Zheng, Han, Wang, & Xiao, 2018), convex optimization (Alexander, Shamir, & Sridharan, 2012; Hazan & Kale, 2014; Shalev-Shwartz, Shamir, Srebro, & KarthikSridharan, 2009; Shalev-Shwartz, Singer, Srebro, & Cotter, 2011; Shamir & Zhang, 2013), reinforcement learning (Pendharkar & Cusatis, 2018; Sutton & Barto, 2018), deep neural networks (Goodfellow, Bengio, & Courville, 2016; Jurgovsky et al., 2018; Nweke, Teh, Al-garadi, & Alo, 2018) or neural networks based expert systems (Gallant, 1993). It stands out as one of the most popular first-order methods for solving optimization problems (Léon & Olivier, 2008; Shalev-Shwartz & Srebro, 2008), particularly for the large-scale optimization problems (in big data) owing to its simplicity and scalability. In particular, it can be used, for instances, to train SVMs (Shalev-shwartz, Singer, & Srebro, 2007; Theodoridis, 2015; Zhou & Jiang, 2018) or convolutional neural networks (Ferreira, Correa, Nonato, & de Mello, 2018), to develop online algorithms for intelligent driving (Cheng, Chen, Cheng, & Zheng, 2017), or to search the optimum solution in non-convex scenarios (Gan, Cao, Wu, & Chen, 2018).

In this paper, we focus on SGD for convex programming problems. Formally, consider the following problem of minimizing an objective defined over a closed convex domain $\mathcal{W}$:

$$\min_{w\in\mathcal{W}} F(w) = \mathbb{E}_Z f(w; z),$$

where for any $z \sim Z$, $f(w; z)$ is a convex function of w. The goal is to find a w to minimize $F(w)$ given i.i.d. samples $z_1, z_2, \ldots$ drawn independently form an unknown distribution Z. SGD sequentially queries the gradient (or more generally, the subgradient) oracle and updates w_t iteratively as follows

$$w_{t+1} = \Pi_{\mathcal{W}}(w_t - \gamma_t g_t),$$

where $\Pi_{\mathcal{W}}$ is the orthogonal projection on $\mathcal{W}$, γ_t is the learning rate at step t and $\mathbb{E}g_t$ is a subgradient of F at w_t. Initially, we require $w_1 \in \mathcal{W}$ and throughout the paper we will simply let $w_1 = 0$. After a number of (e.g., T) steps, SGD produces an approximate solution $\hat{w}$, e.g., by setting

$$\hat{w} = \bar{w}_T = \frac{1}{T}\sum_{t=1}^{T} w_t.$$

The rate of convergence of $\mathbb{E}[F(\hat{w})] - \min_{w\in\mathcal{W}} F(w)$ is often adopted as a measure of performance, where the expectation is taken over the randomness in the gradient oracle. It can be expected that the number of calls (T) limits the precision of the approximate solution and the convergence rate of SGD.

To stabilize the computation, or to avoid overfitting or ill-conditioning or to obtain sparse solutions, regularization techniques have been widely discussed (Ferreira et al., 2018; Goodfellow et al., 2016; Hastie, Tibshirani, & Wainwright, 2015; Richhariya & Tanveer, 2018; Theodoridis, 2015). A typical way to regularize is via introducing an extra term called *Regularizer*, as adopted in SVMs (Jia-Zhi DU, WU, DONG, & ZUO, 2018; Jiang & He, 2012; Jiang & Wang, 2010; Shalev-Shwartz et al., 2007) or regularized regression (Ahn, Byun, Oh, & Kim, 2012; Hastie et al., 2015; Hoerl & Kennard, 1970; Hussein, Elgendi, Wang, & Ward, 2018; Tibshirani, 1996). Specifically, in this case f is a combination of both the loss function and the regularizer, i.e.,

$$f(w; z) = \ell(\langle w, \phi(z)\rangle, z) + R(w),$$

where ℓ is a chosen convex loss and R the regularizer (e.g., $R(\cdot) = \|\cdot\|_2^2$ for the classic ℓ_2-SVM or ridge regression). In this case, the loss function is regularized before applying SGD, and the performance of SGD is then studied for f. Intuitively, in this case, f, the to-be-optimized objective, deviates from the true objection ℓ that should be optimized. This casts doubts on how the output from running SGD on f will perform on optimizing $\mathcal{L}(w) = \mathbb{E}_Z[\ell(\langle w, \phi(z)\rangle, z))]$.

In this paper we revisit the matter of l_2 regularization for SGD, to investigate whether or when an explicit regularizer is necessary to obtain a desirable performance on $\mathcal{L}(w)$. In particular, the SGD algorithm using the gradient of $\mathcal{L}$ (without an explicit regularizer) and the SGD algorithm using the gradient of F (l_2 regularized) are closely studied, and they are denoted respectively by SGDE and SGDER. Further, based on above investigation, we introduce a simple (but with a constant advantage over standard SGD) stochastic averaging method which uses the accumulated stochastic gradient of the loss without any explicit regularizer (denoted by ASG).

The main contributions of this paper summarize as follows.

- We show that in the sense of the upper bound of the convergence rate on the true loss ℓ, SGDE without an explicit regularizer has a constant factor, e.g., of $1/\sqrt{2}$ advantage over SGDER which is l_2 regularized. This result may be explained by observing that there is equivalently a "regularizer" for SGD in the case where a regularizer is not explicitly added.
- When implementing SGDE, the standard analysis, as presented in Zinkevich (2003) or Shamir and Zhang (2013) may not be applicable when there is no projection step performed. In practice the cost of projection steps may not be ignored and it is especially true for large-scale problems in big data. We present a simple algorithm using the accumulated stochastic gradient of the expected loss without regularizer (ASG) to avoid this issue. The algorithm further shows a provable constant factor, e.g., of 1/2 advantage over standard SGD algorithms in the sense of the upper bound.
- Our results provide insights on using SGD algorithms for big data applications, e.g., to accelerate the training of different kinds of SVMs (Jiang & He, 2012; Jiang & Wang, 2010; Richhariya & Tanveer, 2018; Zhou & Jiang, 2018) or regularized regression (Ahn et al., 2012; Hoerl & Kennard, 1970; Theodoridis, 2015) in large-scale, or to improve the performance of online learning algorithms (Cheng et al., 2017; Xiang, Zhang, Gu, & Cai, 2018). In particular, when the iteration number T can be pre-specified, SGDE (without an explicit regularizer) can give us a decent performance on the expected loss; when T cannot be specified in advance as often occurs in online learning scenarios (He, Kwok, Zhu, & Liu, 2017), ASG can improve the performance of standard SGD algorithms.
- For SGD algorithms, the scheme using the last output as the solution may perform with undesirable fluctuations and the scheme that simply takes the average of the output of each iteration also does not guarantee a performance improvement. The schemes that place flexible weights on the latest output can give a better trade-off, which suggests that, for SGD algorithms in application, a tunable averaging scheme may improve the performance.

The remainder of the paper is organized as follows. Section 2 discusses the related work. Section 3 provides a basic review on SGD using the regularized and unregularized expected loss, and some intuitive ways to understand averaging schemes that are helpful for SGD algorithms. Section 4 analyzes and compares the theoretical aspects of SGD algorithms in cases using the gradient of the regularized and unregularized expected loss, and further introduces a static learning step size for SGD in case when T can be pre-specified. The method using the accumulated stochastic gradient (ASG) is presented and analyzed in Section 5. Experimental results are reported in Section 6, and the last section gives some concluding remarks. The proofs of the main results are included in the Appendix A.