Variance reduced Stochastic Gradient Descent

Harkirat S. Behl 13286

1 Introduction

Stochastic Gradient descent is at the heart of most optimization algorithms these days. The most common application is the training of Deep Neural Networks. Stochastic gradient descent was introduced as an improvement over the traditional gradient descent approach, because it is very cheap as it needs to take gradient with respect to just one data point in one iteration. We look at it in more detail in Sec.3. Stochastic gradient descent has slow convergence asymptotically due to the inherent variance [DBL14]. In this paper we look at imporvements over SGD, namely SAG[SLB17] and SVRG[JZ13], which try to reduce this variance of SGD. We first develop the problem and some background in Sec.2. In Sec.3, we look into Gradient descent and Stochastic Gradient Descent. In Sec.4, we look at the SAG algorithm and in Sec.5 at the SVRG algorithm, along with its convergence analysis.

2 Background

2.1 Problem Formulation: Supervised Machine Learning

Parametric supervised machine learning problem can be seen as the following empirical risk minimization problem:
Data: n training examples $(x_i, y_i), i = 1, ..., N$
Prediction function: $h(x, \theta)$ parameterised by $\theta \in R^d$
Empirical regularized risk minimization:

$$\min_{\theta \in R^d} \frac{1}{n} \sum_{i=1}^{n} \{L(y_i, h(x_i, \theta)) + \lambda \Omega(\theta)\} = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) = g(\theta) \tag{1}$$

Optimization: Finding θ that minimizes the empirical regularized risk.

2.2 Definitions: Smoothness and strong convexity

Definition 1. Lipschitz smooth:
A function $f : R^d \to R$ is Lipschitz smooth if its derivatives are Lipschitz continous with constant L :

$$|| \bigtriangledown f(y) - \bigtriangledown f(x)|| \leq L||y - x|| \forall x, y \in R^d \tag{2}$$

A function $f : R^d \to R$ is L-smooth iff it is twice differentiable and

$$\bigtriangledown^2 f(x) \preceq LI \forall x \in R^d \tag{3}$$

Definition 2. μ-strongly convex:
A convex function $f(x)$ is μ-strongly convex if there exists a $\mu > 0$ s.t $\forall \alpha \in [0, 1]$, it holds that:

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha) f(y) - \frac{1}{2} \alpha (1 - \alpha) \mu ||x - y||^2 \tag{4}$$

When f(x) is differentiable, this is equivalent to:

$$f(y) \geq f(x) + < \bigtriangledown f(x), y - x > + \frac{\mu}{2} ||y - x||^2 \tag{5}$$

A twice differentiable function $f : R^d \rightarrow R$ is μ-strongly convex iff

$$\bigtriangledown^2 f(x) \succeq \mu I \forall x \in R^d \tag{6}$$

Condition number is defined as $\kappa = \frac{L}{\mu} \geq 1$

2.3 Convexity in Finite Sum Problems: Supervised Machine Learning

The un-regularized optimization problem is:

$$\min_{\theta} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) = \frac{1}{n} \sum_{i=1}^{N} \{L(y_i, h(x_i, \theta))\} \tag{7}$$

The above problem in Eq.7 is convex (Case 1) when:

1. each $f_i(\theta)$ is convex:
   Convex loss and linear predictions $h(x, \theta)$ = $\theta^T \Phi(x)$
2. each $f_i(\theta)$ is L-smooth:
   Smooth loss and smooth prediction function $h(x_i, \theta)$

It is strongly convex (Case 2) if along with the above conditions, this additional condition is also satisfied:

1. $g(\theta)$ is strongly convex:
   Strongly convex loss and linear predictions $h(x, \theta)$ = $\theta^T \Phi(x)$

3 Gradient descent and stochastic gradient descent

3.1 Gradient descent

Assuming that Case 1 conditions hold:

$$\theta^t = \theta^{t-1} - \gamma_t \bigtriangledown g(\theta^{t-1}) = \theta^{t-1} - \frac{\gamma_t}{n} \sum_{i=1}^{N} \bigtriangledown f_i(\theta^{t-1}) \tag{8}$$

The convergence rate is $O(1/t)$ for Case 1, i.e convex functions. And the convergence rate is $O(e^{-t/\kappa})$ linear for Case 2, i.e strongly-convex, and the problem complexity is $O(nd * \kappa \log 1/\epsilon)$ [BS17]. We just use the results for convergence for gradient descent and stochastic gradient descent in this paper, and do not look into thier convergence analysis as the main focus of this paper is the variance reduction method.

3.2 Stochastic Gradient descent

At every iteration $t = 1, 2, 3....$, a random i_t is drawn from $i = \{1, ..., n\}$:

$$\theta^t = \theta^{t-1} - \gamma_t \bigtriangledown f_{i_t}(\theta^{t-1}) \tag{9}$$

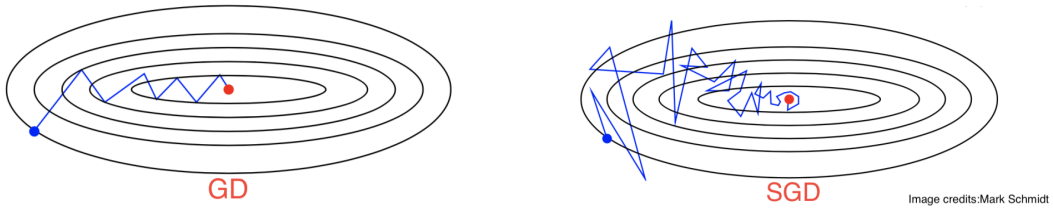
For Case 2, the convergence rate in $O(\kappa/t)$. And the problem complexity is independent of n [BS17]. The more general version of SGD can be written as:

$$\theta^t = \theta^{t-1} - \gamma_t g(\theta^{t-1}, \xi_t) \tag{10}$$

here ξ_t is a random variable, which might depend on θ^{t-1}, such that the expectation $E[g(\theta^{t-1}, \xi_t)|\theta^{t-1}] = \bigtriangledown g(\theta^{t-1})$. This randomness introduces large variance. And a large $g(\theta^{t-1}, \xi_t)$ can slow down the convergence. In Sec5 we look at a way to get rid of this problem, i.e variance reduction.

3.3 Difference between stochastic and deterministic gradient descent methods.

- In every iteration Gradient descent requires evaluation of N derivatives, which is expensive. SGD overcomes this problem as only single gradient computation is needed in every iteration.
- Complexity (number of iterations or running time) for Gradient descent is linear in n : $O(d * n\kappa * \log 1/\epsilon)$, whereas for SGD it is independent of n.
- The convergence rate is linear or exponential for GD: $O(e^{-t/\kappa})$, whereas for SGD it is $O(\kappa/t)$.
- The difference is also illustrated in Fig:3.3. We can see that GD converged in lesser number of iterations, but GD requires much more computation in each iteration(evaluation of N derivatives). This motivates the SAG algorithm in Sec.4.



3.4 Limitations of SGD

Stochastic gradient descent has large variance because of the randomness in the algorithm. It has **slow convergence asymptotically due to this inherent variance**.

4 SAG: Stochastic average gradient [SLB17]

At every iteration $t = 1, 2, 3....$, a random i_t is drawn from $i = \{1, ..., n\}$:

$$\theta^t = \theta^{t-1} - \frac{\gamma}{n} \sum_{i=1}^{n} d_i^t \tag{11a}$$

where

$$d_i^t = \begin{cases} \bigtriangledown f_i(\theta^{t-1}) & \text{if } i = i_t \\ d_i^{t-1} & \text{otherwise} \end{cases} \tag{11b}$$

This requires to store the gradients of all the functions $f_i, i = \{1, ..., n\}$, which takes extra memory, gradient $\in R^d$.

For Case 2, SAG also **has linear or exponential convergence rate**. And the complexity is $O(d * (\kappa + n) * \log 1/\epsilon)$ [SLB17], thus **complexity is linear in** d.

Thus it is able to overcome the limitation of SGD of slow convergence, but **needs extra storage space**.

SAGA [DBL14] The update equation for SAGA (an improvement over SAG) in the variance reduction form can be written as:

$$\theta^t = \theta^{t-1} - \gamma[\frac{1}{n} \sum_{i=1}^{N} y_{i_t}^{t-1} + (\bigtriangledown f_{i_t}(\theta^{t-1}) - y_{i_t}^{t-1})] \tag{12}$$

5 SVRG: Stochastic Variance Reduced Gradient Descent[JZ13]

5.1 Variance reduction

Variance reduction is a technique which is used to reduce the variance of a random variable X by using another random variable Y, which is positively correlated with X. A new variable Z_α is defined

as:

$$Z_\alpha = \alpha(X - Y) + E[Y] \tag{13}$$

It can be seen that expectation of Z_α is $E[Z_\alpha] = \alpha E[X] + (1 - \alpha)E[Y]$. And its variance is $var(Z_\alpha) = \alpha^2(var(X) + var(Y) - 2cov(X, Y))$

5.2 SVRG Procedure

The update equation for θ^t is:

$$\theta^t = \theta^{t-1} - \gamma[\bigtriangledown g(\tilde{\theta}) + (\bigtriangledown f_{i_t}(\theta^{t-1}) - \bigtriangledown f_{i_t}(\tilde{\theta}))] \tag{14}$$

and $\tilde{\theta}$ is updated after every m iterations of Eq. 14

A well formulated algorithm is as follows:

Initialize $\tilde{\theta} \in R^d$;
for *T epochs* **do**
  Compute $\bigtriangledown g(\tilde{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \bigtriangledown f_i(\tilde{\theta})$;
  Initialize $\theta_0 = \tilde{\theta}$;
  for *t = 1 to m* **do**
    $\theta^t = \theta^{t-1} - \gamma[\bigtriangledown g(\tilde{\theta}) + (\bigtriangledown f_{i_t}(\theta^{t-1}) - \bigtriangledown f_{i_t}(\tilde{\theta}))]$;
  end
  Update $\tilde{\theta} = \theta_m$;
end
Output: $\tilde{\theta}$

Algorithm 1: SVRG

5.3 SVRG as variance reduction

SVRG is a form of variance reduction which is discussed in Sec.5.1. The gradient in Eq.14 can be seen as a form of variance reduction. It can be obtained by substituting $X = \bigtriangledown f_{i_t}(\theta^{t-1})$, $Y = \bigtriangledown f_{i_t}(\widetilde{\theta})$, $\alpha = 1$ in Eq.13.

5.4 Convergence of SVRG

Therorem: Assume all f_i are convex and smooth, and $g(\theta)$ is strongly convex with $\gamma > 0$. Let us assume that $\theta^* = argmin_\theta g(\theta)$. The convergence in expectation for SVRG is:

$$E[g(\tilde{\theta^s})] \leq E[g(\theta^*)] + \alpha^s E[g(\tilde{\theta^0}) - g(\theta^*)] \tag{15}$$

Proof: For any i, consider

$$p_i(\theta) = f_i(\theta) - f_i(\theta^*) - \bigtriangledown f_i(\theta^*)^T(\theta - \theta^*) \tag{16}$$

where $p_i(\theta^*) = min(p_i(\theta))$, because $\bigtriangledown p_i(\theta^*) = 0$, this gives:

$$\begin{aligned} 0 &= p_i(\theta^*) \\ &\leq min_\eta[p_i(\theta - \eta \bigtriangledown p_i(\theta)] \\ &\leq min_\eta[p_i(\theta) - \eta|| \bigtriangledown p_i(\theta)||_2^2 + 0.5L\eta^2|| \bigtriangledown p_i(\theta)||_2^2] \\ &= p_i(\theta) - \frac{1}{2L}|| \bigtriangledown p_i(\theta)||_2^2 \end{aligned}$$

that is:

$$|| \bigtriangledown f_i(\theta) - \bigtriangledown f_i(\theta^*)||_2^2 \leq 2L[f_i(\theta) - f_i(\theta^*) - \bigtriangledown f_i(\theta^*)^T(\theta - \theta^*)]$$

Summing the above inequality over $i = 1, ..., n.$, and using $\bigtriangledown g(\theta^*) = 0$:

$$\frac{1}{n}\sum_{i=1}^{n} || \bigtriangledown f_i(\theta) - \bigtriangledown f_i(\theta^*)||_2^2 \leq 2L[g(\theta) - g(\theta^*)] \tag{17}$$

We will use this result.

Let $v_t = \bigtriangledown f_{i_t}(\theta^{t-1}) - \bigtriangledown f_{i_t}(\tilde{\theta}) + \tilde{\mu}$. Conditioning on θ^{t-1}, we will take expectation of v_t wrt i_t:

$$
\begin{aligned}
E||v_t||_2^2 &\leq 2E||\bigtriangledown f_{i_t}(\theta^{t-1}) - \bigtriangledown f_{i_t}(\theta^*)||_2^2 + 2E||[\bigtriangledown f_{i_t}(\tilde{\theta}) - \bigtriangledown f_{i_t}(\theta^*)] - \bigtriangledown g(\tilde{\theta})||_2^2 \\
&\qquad\qquad \text{using } ||a+b||_2^2 \leq 2||a||_2^2 + 2||b||_2^2 \text{ and } \tilde{\mu} = \bigtriangledown g(\tilde{\theta}) \\
&= 2E||\bigtriangledown f_{i_t}(\theta^{t-1}) - \bigtriangledown f_{i_t}(\theta^*)||_2^2 + 2E||[\bigtriangledown f_{i_t}(\tilde{\theta}) - \bigtriangledown f_{i_t}(\theta^*)] - E[\bigtriangledown f_{i_t}(\tilde{\theta}) - \bigtriangledown f_{i_t}(\theta^*)]||_2^2 \\
&\leq 2E||\bigtriangledown f_{i_t}(\theta^{t-1}) - \bigtriangledown f_{i_t}(\theta^*)||_2^2 + 2E||\bigtriangledown f_{i_t}(\tilde{\theta}) - \bigtriangledown f_{i_t}(\theta^*)||_2^2 \\
&\qquad\qquad \text{using } E||\xi - E\xi||_2^2 = E||\xi||_2^2 - ||E\xi||_2^2 \leq E||\xi||_2^2 \text{ for any random vector } \xi \\
&\leq 4L[g(\theta^{t-1}) - g(\theta^*) + g(\tilde{\theta}) + g(\theta^*)] \qquad \text{using Eq.17}
\end{aligned}
$$

We conditioned on w_{t-1}, we get $Ev_t = \bigtriangledown g(\theta^{t-1})$, which gives:

$$
\begin{aligned}
E||\theta^t - \theta^*||_2^2 &= ||\theta^{t-1} - \theta^*||_2^2 - 2\eta(\theta^{t-1} - \theta^*)^T Ev_t + \eta^2 E||v_t||_2^2 \\
&\leq ||\theta^{t-1} - \theta^*||_2^2 - 2\eta(\theta^{t-1} - \theta^*)^T \bigtriangledown g(\theta^{t-1}) + \eta^2 4L[g(\theta^{t-1}) - g(\theta^*) + g(\tilde{\theta}) + g(\theta^*)] \\
&\qquad\qquad \text{using result of previous inequality} \\
&\leq ||\theta^{t-1} - \theta^*||_2^2 - 2\eta[g(\theta^{t-1}) - g(\theta^*)] + 4\eta^2 L[g(\theta^{t-1}) - g(\theta^*) + g(\tilde{\theta}) + g(\theta^*)] \\
&\qquad\qquad \text{using convexity of g}(\theta) \\
&= ||\theta^{t-1} - \theta^*||_2^2 - 2\eta(1 - 2L\eta)[g(\theta^{t-1}) - g(\theta^*)] + 4\eta^2 L[g(\tilde{\theta}) + g(\theta^*)]
\end{aligned}
$$

Let us consider a particular state s, s.t $\tilde{w} = \tilde{w}_{s-1}$. $\tilde{w}_s$ is selected after all the inner updates hace been completed. Summing the previous inequality for $t = 1, ..., m,$, i.e over all iterations of the inner loop in the algorithm:

$$
\begin{aligned}
E||\theta^m - \theta^*||_2^2 + 2\eta(1 - 2L\eta)mE[g(\tilde{\theta^s}) - g(\theta^*)] &\leq E||\theta^0 - \theta^*||_2^2 + 4Lm\eta^2 E[g(\tilde{\theta}) - g(\theta^*)] \\
&= E||\tilde{\theta} - \theta^*||_2^2 + 4Lm\eta^2 E[g(\tilde{\theta}) - g(\theta^*)] \\
&\leq \frac{2}{\gamma} E[g(\tilde{\theta}) - g(\theta^*)] + 4Lm\eta^2 E[g(\tilde{\theta}) - g(\theta^*)] \\
&\qquad\qquad \text{using strong convexity Sec.2.2} \\
&= 2(\gamma^{-1} + 2Lm\eta^2) E[g(\tilde{\theta^0}) - g(\theta^*)]
\end{aligned}
$$

This gives:

$$
E[g(\tilde{\theta^s}) - g(\theta^*)] \leq \Big[\frac{1}{\gamma\eta(1 - 2L\eta)m} + \frac{2L\eta}{1 - 2L\eta}\Big] E[g(\tilde{\theta^{s-1}}) - g(\theta^*)] \tag{18}
$$

Which implies $E[g(\tilde{\theta^s}) - g(\theta^*)] \leq \alpha^s E[g(\tilde{\theta^0}) - g(\theta^*)]$.

Hence proved.

5.5 SVRG overcomes the limitations of SGD and SAG

- **SVRG does not need to store gradients, unlike SAG**.
- It has **linear convergence rate**, same as SAG.
- Unlike SAG, it is more easily applicable to complex problems.

References

[JZ13] Rie Johnson and Tong Zhang. "Accelerating Stochastic Gradient Descent using Predictive Variance Reduction". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 315–323. URL: `http://papers.nips.cc/paper/4937-accelerating-stochastic-gradient-descent-using-predictive-variance-reduction.pdf`.

[DBL14] Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. “SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives”. In: *CoRR* abs/1407.0202 (2014). arXiv: 1407.0202. URL: http://arxiv.org/abs/1407.0202.

[BS17] Francis Bach and Mark Schmidt. *SIAM Conference on Optimization mini-tutorial on "Stochastic Variance-Reduced Optimization for Machine Learning"*. http://www.di.ens.fr/~fbach/fbach_tutorial_siopt_2017.pdf. Accessed: 2018-04-03. 2017.

[SLB17] Mark Schmidt, Nicolas Le Roux, and Francis Bach. “Minimizing Finite Sums with the Stochastic Average Gradient”. In: *Math. Program.* 162.1-2 (Mar. 2017), pp. 83–112. ISSN: 0025-5610. DOI: 10.1007/s10107-016-1030-6. URL: https://doi.org/10.1007/s10107-016-1030-6.