1 INTUITIONS

1.1 CONNECTION TO VARIANCE REDUCTION APPROACH

A lot of recent work has been devoted to the study of variance reduction techniques in convex optimization, including Johnson & Zhang (2013); Shalev-Shwartz & Zhang (2013b;a); Xiao & Zhang (2014); Schmidt et al. (2017); Defazio et al. (2014); Mairal (2015); Frostig et al. (2015); Lee et al. (2015); Lei & Jordan (2016); Lin et al. (2015); Allen-Zhu (2017); Bietti & Mairal (2017). These methods are able to achieve linear convergence rates for smooth strongly-convex optimization problems, which significantly improve the sub-linear rate of vanilla SGD. Extension of variance reduction approaches to general non-convex problems has also been studied in Allen-Zhu & Hazan (2016); Reddi et al. (2016). Despite these theoretical progress, variance reduction technique does not appear to be helpful for training deep neural network Defazio & Bottou (2018).

Our proposed method could be viewed as a delayed variant of SVRG in the decentralized setting. The update (1) in our method mimic the update of vanilla SVRG, with the difference that the average gradient is obtained across machine base on local parameters. The variance reduction technique empirically stabilized the asynchronous gradients and is able to achieve state-of-the-art performance in modern deep learning setting. Moreover, we obtain close linear speed up with respect to the number of machine, which is the essential goal of distributed optimization.

2 THEORETICAL JUSTIFICATION

Our proof closely follows the analysis in Yu et al. (2019). We consider the following non-convex optimization problem

$$\min_{w} f(w) = \frac{1}{J}\sum_{j=1}^{J} f_j(w),$$

where J denotes the number of workers. Moreover, each $f_j = \mathbb{E}_{\zeta_j}[F_j(w, \zeta_j)]$. In the empirical minimization setting, ζ_j corresponds to a mini-batch sampling strategy and f_j could again be expressed as a finite sum $f_j = \frac{1}{K_j}\sum f_{k_j}$. Throughout this paper, we assume the optimization satisfies the following assumptions.

Assumption 1 (L-smooth)**.** *Each function $f_j(x)$ is L-smooth,* i.e. *for all $x, y \in \mathbb{R}^d$ and $i \in [1, J]$,*

$$||\nabla f_j(x) - \nabla f_j(y)|| \leq L||x - y||.$$

Assumption 2 (Bounded gradients and variances)**.** *There exists a positive constant G and σ such that*

$$\mathbb{E}_{\zeta_j}||\nabla F_j(w; \zeta_i)||^2 \leq G^2, \forall w, \forall j$$

$$\mathbb{E}_{\zeta_j}||\nabla F_j(w; \zeta_j) - \nabla f_j(w)||^2 \leq \sigma^2, \forall w, \forall j$$

2.1 PROOF OF THE CONVERGENCE OF DELAYED UPDATE ALGORITHM

Delayed Update Algorithm

1. Initialize w_0 and initialize each worker with $w_{0,j} = w_0$ for $j \in [1, J]$.
2. **For** iteration $n = 0, 1, \cdots$

   On each local worker j, sample a stochastic gradient $\nabla w_{(n-1,j)}$ and update:
   - If $n < t$, update with standard SGD rule:
   $$w_{(n,j)} = w_{(n-1,j)} - \gamma \nabla w_{(n-1,j)}$$
   - Else:
   $$w_{(n,j)} = w_{(n-1,j)} - \gamma(\nabla w_{(n-1,j)} + \overline{\nabla w_{n-t}} - \nabla w_{(n-t,j)}) \tag{1}$$
   where
   $$\overline{\nabla w_{n-t}} = \frac{1}{J}\sum_{j=1}^{J} \nabla w_{(n-t,j)}$$
3. **Output** $\overline{w_n} = \frac{1}{J}\sum_{j=1}^{J} w_{(n,j)}$

Notably, the delayed update formula can rewritten as

$$w_{(n,j)} = \underbrace{w_0 - \gamma \sum_{i=0}^{n-t} \overline{\nabla w_i}}_{\text{invariant of } j} - \gamma \sum_{i=n-t+1}^{n-1} \nabla w_{(i,j)}, \tag{Delayed}$$

where the first quantity is commonly shared among all the workers and the second term corresponds to local updates.

Lemma 2.1. *Let $(w_{(i,j)})$ be the sequence generated on the j-th worker according to Delayed Update Algorithm, then for any $j_0 \in [1, J]$*

$$\mathbb{E}\left[\|\overline{w_n} - w_{(n,j_0)}\|^2\right] \leq 4\gamma^2 t^2 G^2$$

Proof. From the delayed update formula Delayed, we have,

$$\begin{aligned}
&\mathbb{E}\left[\|\overline{w_n} - w_{(n,j_0)}\|^2\right] \\
=& \gamma^2 \mathbb{E}\left[\left\|\frac{1}{J}\sum_{j=1}^{J}\sum_{i=n-t+1}^{n-1} \nabla w_{(i,j)} - \sum_{i=n-t+1}^{n-1} \nabla w_{(i,j_0)}\right\|^2\right] \\
\leq& 2\gamma^2 \mathbb{E}\left[\left\|\frac{1}{J}\sum_{j=1}^{J}\sum_{i=n-t+1}^{n-1} \nabla w_{(i,j)}\right\|^2\right] + \mathbb{E}\left[\|\sum_{i=n-t+1}^{n-1} \nabla w_{(i,j_0)}\|^2\right] \\
\leq& \frac{2\gamma^2 t}{J}\sum_{j=1}^{J}\sum_{i=n-t+1}^{n-1} \mathbb{E}\left[\left\|\nabla w_{(i,j)}\right\|^2\right] + 2\gamma^2 t \sum_{i=n-t+1}^{n-1} \mathbb{E}\left[\|\nabla w_{(i,j_0)}\|^2\right] \\
\leq& 4\gamma^2 t^2 G^2.
\end{aligned}$$

□

Theorem 2.2. *Consider the sequence generated by the Delayed Algorithm, then*

$$\frac{1}{N}\sum_{n=0}^{N-1}\mathbb{E}[\|\nabla f(\overline{w_n})\|^2] \leq \frac{2}{\gamma N}(f(w_0)-f^*) + 4\gamma^2 t^2 G^2 L^2 + \frac{L}{J}\gamma\sigma^2.$$

Proof. By smoothness of f, we have

$$\mathbb{E}[f(\overline{w_n})] \leq \mathbb{E}[f(\overline{w_{n-1}}) + \langle \nabla f(\overline{w_{n-1}}), \overline{w_n} - \overline{w_{n-1}}\rangle + \frac{L}{2}\|\overline{w_n} - \overline{w_{n-1}}\|^2]$$

Note that

$$\overline{w_n} = \overline{w_{n-1}} - \frac{\gamma}{J}\sum_{j=1}^{J}\nabla w_{(n-1,j)}.$$

We have

$$\begin{aligned}
&\mathbb{E}[\|\overline{w_n} - \overline{w_{n-1}}\|^2] \\
=&\gamma^2\mathbb{E}[\|\frac{1}{J}\sum_{j=1}^{J}\nabla w_{(n-1,j)}\|^2] \\
=&\gamma^2\mathbb{E}[\|\frac{1}{J}\sum_{j=1}^{J}(\nabla w_{(n-1,j)} - \nabla f_j(w_{(n-1,j)})\|^2] + \gamma^2\mathbb{E}[\|\frac{1}{J}\sum_{j=1}^{J}\nabla f_j(w_{(n-1,j)})\|^2] \\
=&\frac{\gamma^2}{J^2}\sum_{j=1}^{J}\mathbb{E}[\|(\nabla w_{(n-1,j)} - \nabla f_j(w_{(n-1,j)})\|^2] + \gamma^2\mathbb{E}[\|\frac{1}{J}\sum_{j=1}^{J}\nabla f_j(w_{(n-1,j)})\|^2] \\
\leq&\frac{\gamma^2\sigma^2}{J} + \gamma^2\mathbb{E}[\|\frac{1}{J}\sum_{j=1}^{J}\nabla f_j(w_{(n-1,j)})\|^2].
\end{aligned}$$

Moreover,

$$\begin{aligned}
&\mathbb{E}[\langle \nabla f(\overline{w_{n-1}}), \overline{w_n} - \overline{w_{n-1}}] \\
=&-\gamma\mathbb{E}[\langle \nabla f(\overline{w_{n-1}}), \frac{1}{J}\sum_{j=1}^{J}\nabla w_{(n-1,j)}] \\
=&-\gamma\mathbb{E}[\langle \nabla f(\overline{w_{n-1}}), \frac{1}{J}\sum_{j=1}^{J}\nabla f_j(w_{(n-1,j)}] \\
=&-\frac{\gamma}{2}\mathbb{E}[\|\nabla f(\overline{w_{n-1}})\|^2] + \frac{\gamma}{2}\mathbb{E}[\|\nabla f(\overline{w_{n-1}}) - \frac{1}{J}\sum_{j=1}^{J}\nabla f_j(w_{(n-1,j)}\|^2] - \frac{\gamma}{2}\mathbb{E}[\|\frac{1}{J}\sum_{j=1}^{J}\nabla f_j(w_{(n-1,j)})\|^2]
\end{aligned}$$

Applying the lemma, we have

$$\begin{aligned}
&\mathbb{E}[\|\nabla f(\overline{w_{n-1}}) - \frac{1}{J}\sum_{j=1}^{J}\nabla f_j(w_{(n-1,j)}\|^2] \\
=&\mathbb{E}[\|\frac{1}{J}\sum_{j=1}^{J}(\nabla f_j(\overline{w_{n-1}}) - \nabla f_j(w_{(n-1,j)})\|^2] \\
\leq&\frac{1}{J}\sum_{j=1}^{J}\mathbb{E}[\|(\nabla f_j(\overline{w_{n-1}}) - \nabla f_j(w_{(n-1,j)})\|^2] \\
\leq&\frac{L^2}{J}\sum_{j=1}^{J}\|\overline{w_{n-1}} - w_{(n-1,j)}\|^2 \\
\leq&4\gamma^2 L^2 t^2 G^2.
\end{aligned}$$

Thus, combining the above inequalities together with $\gamma \leq \frac{1}{L}$ yields

$$\begin{aligned}
&\mathbb{E}[f(\overline{w_n})] - \mathbb{E}[f(\overline{w_{n-1}})] \\
\leq &\frac{\gamma L^2 - \gamma}{2}\mathbb{E}[\|\frac{1}{J}\sum_{j=1}^{J}\nabla f_j(w_{(n-1,j)})\|^2] - \frac{\gamma}{2}\mathbb{E}[\|\nabla f(\overline{w_{n-1}})\|^2] + 2\gamma^3 L^2 t^2 G^2 + \frac{L}{2J}\gamma^2\sigma^2 \\
\leq &-\frac{\gamma}{2}\mathbb{E}[\|\nabla f(\overline{w_{n-1}})\|^2] + 2\gamma^3 L^2 t^2 G^2 + \frac{L}{2J}\gamma^2\sigma^2.
\end{aligned}$$

Rearrange the terms yields

$$\mathbb{E}[\|\nabla f(\overline{w_{n-1}})\|^2] \leq \frac{2}{\gamma}(\mathbb{E}[f(\overline{w_{n-1}})] - \mathbb{E}[f(\overline{w_n})]) + 4\gamma^2 L^2 t^2 G^2 + \frac{L}{J}\gamma\sigma^2$$

Telescoping from $n = 1, ..N$ yields

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[\|\nabla f(\overline{w_{n-1}})\|^2] \leq \frac{2}{\gamma}(\mathbb{E}[f(\overline{w_0})] - \mathbb{E}[f(\overline{w_n})]) + 4\gamma^2 L^2 t^2 G^2 + \frac{L}{J}\gamma\sigma^2$$

□

Corollary 2.2.1. *When the number of iteration N surpass the number of machines J, let the stepsize $\gamma = \frac{\sqrt{J}}{L\sqrt{N}}$, yields*

$$\begin{aligned}
\frac{1}{N}\sum_{n=0}^{N-1}\mathbb{E}[\|\nabla f(\overline{w_n})\|^2] &\leq \frac{2L}{\sqrt{JN}}(f(w_0) - f^* + \sigma^2) + \frac{4t^2G^2J}{N}. \\
&= O(\frac{1}{\sqrt{JN}}) + O(\frac{t^2J}{N})
\end{aligned}$$

When iterations $N \geq O(t^4J^3)$, the first term dominates and delayed update demonstrates the same convergence speed as original SSGD $O(1/\sqrt{JN})$.

2.2 Proof of the convergence of Temporally Sparse Update Algorithm

Temporally Sparse Update Algorithm

1. Initialize w_0 and initialize each worker with $w_{0,j} = w_0$ for $j \in [1, J]$.
2. **For** iteration $n = 0, 1, \cdots$

   On each local worker j, sample a stochastic gradient $\nabla w_{(n-1,j)}$ and update:
   - If $n \pmod d \neq 0$, update with standard (local) SGD rule:

   $$w_{(n,j)} = w_{(n-1,j)} - \gamma\nabla w_{(n-1,j)}$$

   - Else:

   $$w_{(n,j)} = w_{(n-1,j)} - \gamma[\nabla w_{(n-1,j)} + \sum_{k=n-d}^{n-1}(\overline{\nabla w_{n-d+k}} - \nabla w_{(n-d+k,j)})] \quad (2)$$

   where

   $$\overline{\nabla w_{n-d+k}} = \frac{1}{J}\sum_{j=1}^{J}\nabla w_{(n-d+k,j)}$$

3. **Output** $\overline{w_n} = \frac{1}{J}\sum_{j=1}^{J} w_{(n,j)}$

Notably, the temporally sparse update formula can rewritten as

$$w_{(n,j)} = \underbrace{w_0 - \gamma \sum_{i=0}^{d\lfloor \frac{n}{d} \rfloor} \overline{\nabla w_i}}_{\text{invariant of } j} - \gamma \sum_{i=d\lfloor \frac{n}{d} \rfloor+1}^{n-1} \nabla w_{(i,j)}, \tag{TS}$$

where the first quantity is commonly shared among all the workers and the second term corresponds to local updates.

Lemma 2.3. *Let $(w_{(i,j)})$ be the sequence generated on the j-th worker according to Temporally Sparse Update Algorithm, then for any $j_0 \in [1, J]$*

$$\mathbb{E}\left[\|\overline{w_n} - w_{(n,j_0)}\|^2\right] \leq 4\gamma^2 d^2 G^2$$

Proof. From the update of (TS), we have

$$\begin{aligned}
&\mathbb{E}\left[\|\overline{w_n} - w_{(n,j_0)}\|^2\right] \\
=& \gamma^2 \mathbb{E}\left[\left\|\frac{1}{J}\sum_{j=1}^{J}\sum_{i=d\lfloor \frac{n}{d} \rfloor+1}^{n-1} \nabla w_{(i,j)} - \sum_{i=d\lfloor \frac{n}{d} \rfloor+1}^{n-1} \nabla w_{(i,j_0)}\right\|^2\right] \\
\leq& 2\gamma^2 \mathbb{E}\left[\left\|\frac{1}{J}\sum_{j=1}^{J}\sum_{i=d\lfloor \frac{n}{d} \rfloor+1}^{n-1} \nabla w_{(i,j)}\right\|^2\right] + \mathbb{E}\left[\|\sum_{i=d\lfloor \frac{n}{d} \rfloor+1}^{n-1} \nabla w_{(i,j_0)}\|^2\right] \\
\leq& \frac{2\gamma^2 d}{J}\sum_{j=1}^{J}\sum_{i=d\lfloor \frac{n}{d} \rfloor+1}^{n-1} \mathbb{E}\left[\left\|\nabla w_{(i,j)}\right\|^2\right] + 2\gamma^2 d \sum_{i=d\lfloor \frac{n}{d} \rfloor+1}^{n-1} \mathbb{E}\left[\|\nabla w_{(i,j_0)}\|^2\right] \\
\leq& 4\gamma^2 d^2 G^2.
\end{aligned}$$

□

Theorem 2.4. *Consider the sequence generated by the Temporally Sparse Update Algorithm, then*

$$\frac{1}{N}\sum_{n=0}^{N-1} \mathbb{E}[\|\nabla f(\overline{w_n})\|^2] \leq \frac{2}{\gamma N}(f(w_0) - f^*) + 4\gamma^2 d^2 G^2 L^2 + \frac{L}{J}\gamma\sigma^2.$$

Proof. Based on Lemma 2.3, the proof follows the exact same schema as the Delayed Update Algorithm, which we omit here. □

Corollary 2.4.1. *When the number of iteration N surpass the number of machines J, let the stepsize $\gamma = \frac{\sqrt{J}}{L\sqrt{N}}$, yields*

$$\begin{aligned}
\frac{1}{N}\sum_{n=0}^{N-1} \mathbb{E}[\|\nabla f(\overline{w_n})\|^2] &\leq \frac{2L}{\sqrt{JN}}(f(w_0) - f^* + \sigma^2) + \frac{4d^2G^2J}{N}. \\
&= O(\frac{1}{\sqrt{JN}}) + O(\frac{d^2 J}{N})
\end{aligned}$$

When iterations $N \geq O(d^4 J^3)$, the first term dominates and delayed update demonstrates the same convergence speed as original SSGD.

REFERENCES

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International conference on machine learning*, pp. 699–707, 2016.

Alberto Bietti and Julien Mairal. Stochastic optimization with variance reduction for infinite datasets with finite sum structure. In *Advances in Neural Information Processing Systems*, pp. 1623–1633, 2017.

Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *arXiv preprint arXiv:1812.04529*, 2018.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pp. 1646–1654, 2014.

Roy Frostig, Rong Ge, Sham M Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *Conference on learning theory*, pp. 728–763, 2015.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pp. 315–323, 2013.

Jason D Lee, Qihang Lin, Tengyu Ma, and Tianbao Yang. Distributed stochastic variance reduced gradient methods and a lower bound for communication complexity. *arXiv preprint arXiv:1507.07595*, 2015.

Lihua Lei and Michael I Jordan. Less than a single pass: Stochastically controlled stochastic gradient method. *arXiv preprint arXiv:1609.03261*, 2016.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in neural information processing systems*, pp. 3384–3392, 2015.

Julien Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pp. 314–323, 2016.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *arXiv preprint arXiv:1309.2375*, 2013a.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013b.

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.