Slow and Stale Gradients Can Win the Race: Error-Runtime Trade-offs in Distributed SGD
Supplement

6 STRONG CONVEXITY

Definition 4 (Strong-Convexity). A function $h(u)$ is defined to be $c$-strongly convex, if the following holds for all $u_1$ and $u_2$ in the domain:

$$h(u_2) \geq h(u_1) + [\nabla h(u_1)]^T (u_2 - u_1) + \frac{c}{2} ||u_2 - u_1||^2.$$ 

For strongly convex functions, the following result holds for all $u$ in the domain of $h(.)$.

$$2c(h(u) - h^*) \leq ||\nabla h(u)||^2. \quad (19)$$

The proof is derived in [Bottou et al., 2016]. For completeness, we give the sketch here.

Proof. Given a particular $u$, let us define the quadratic function as follows:

$$q(u') = h(u) + [\nabla h(u)]^T (u' - u) + \frac{c}{2} ||u' - u||^2$$

Now, $q(u')$ is minimized at $u' = u - \frac{1}{2c} \nabla h(u)$ and the value is $h(u) - \frac{1}{2c} ||\nabla h(u)||^2$. Thus, from the definition of strong convexity we now have,

$$h^* \geq h(u) + [\nabla h(u)]^T (u' - u) + \frac{c}{2} ||u' - u||^2$$

$$\geq h(u) - \frac{1}{2c} ||\nabla h(u)||^2 \quad \text{[minimum value of } q(u')] \quad \blacksquare$$

7 RUNTIME ANALYSIS PROOFS

Here we provide all the proofs and supplementary information for all the results in Section 4.

7.1 Runtime of $K$-sync SGD

Proof of Lemma 3. We assume that the $P$ learners have an i.i.d. computation times. When all the learners start together, and we wait for the first $K$ out of $P$ i.i.d. random variables to finish, the expected computation time for that iteration is $E[X_{K,P}]$, where $X_{K,P}$ denotes the $K$-th statistic of $P$ i.i.d. random variables $X_1, X_2, \ldots, X_P$. Thus, for $J$ iterations, the runtime is given by $J E[X_{K,P}]$.

7.1.1 $K$-th statistic of exponential distributions

Here we give a sketch of why the $K$-th order statistic of $P$ exponentials scales as $\log(P/P - K)$. A detailed derivation can be obtained in [Sheldon, 2002]. Consider $P$ i.i.d. exponential distributions with parameter $\mu$. The minimum $X_{1:P}$ of $P$ independent exponential random variables with parameter $\mu$ is exponential with parameter $P\mu$. Conditional on $X_{1:P}$, the second smallest value $X_{2:P}$ is distributed like the sum of $X_{1:P}$ and an independent exponential random variable with parameter $(P - 1)\mu$. And so on, until the $K$-th smallest value $X_{K:P}$ which is distributed like the sum of $X_{(K-1):P}$ and an independent exponential random variable with parameter $(P - K + 1)\mu$. Thus,

$$X_{K:P} = Y_{P} + Y_{P-1} + \cdots + Y_{P-K+1}$$

where the random variables $Y_i$s are independent and exponential with parameter $i\mu$. Thus,

$$E[X_{K:P}] = \sum_{i=P-K+1}^{P} \frac{1}{i\mu} = \frac{H_P - H_{P-K}}{\mu} \approx \frac{\log \frac{P}{P-K}}{\mu}.$$ 

Here $H_P$ and $H_{P-K}$ denote the $P$-th and $(P - K)$-th harmonic numbers respectively.

For the case where $K = P$, the expectation is given by,

$$E[X_{P:P}] = \frac{1}{\mu} \sum_{i=1}^{P} \frac{1}{i} = \frac{1}{\mu} H_P \approx \frac{1}{\mu} \log P.$$ 

7.2 Runtime of $K$-batch-async SGD

Here we include a discussion on renewal processes for completeness, as a background to the proof of Lemma 4,
which gives the runtime of $K$-batch-async SGD. The familiar reader can skim through this part, and directly proceed to the proof of Lemma 4 in the main paper in Section 4.

**Definition 5 (Renewal Process).** A renewal process is an arrival process where the inter-arrival intervals are positive, independent and identically distributed random variables.

**Lemma 6 (Elementary Renewal Theorem).** [Gallager, 2013, Chapter 5] Let $\{N(t), t > 0\}$ be a renewal counting process denoting the number of renewals in time $t$. Let $E[Z]$ be the mean inter-arrival time. Then,

$$\lim_{t \to \infty} \frac{E[N(t)]}{t} = \frac{1}{E[Z]} \quad \text{(20)}$$

Observe that for asynchronous SGD or $K$-batch-async SGD, every gradient push by a learner to the PS can be thought of as an arrival process. The time between two consecutive pushes by a learner follows the distribution of $X_i$ and is independent as computation time has been assumed to be independent across learners and mini-batches. Thus the inter-arrival intervals are positive, independent and identically distributed and hence, the gradient pushes are a renewal process.

### 7.3 Runtime of $K$-async SGD

**Proof of Lemma 5.** For new-longer-than-used distributions observe that the following holds:

$$\Pr(X_i > u + t | X_i > t) \leq \Pr(X_i > u) \quad \text{(21)}$$

Thus the random variable $X_i - t | X_i > t$ is thus stochastically dominated by $X_i$. Now let us assume we want to compute the expected computation time of one iteration of $K$-async starting at time instant $t_0$. Let us also assume that the learners last read their parameter values at time instants $t_1, t_2, \ldots, t_P$ respectively where any $K$ of these $t_1, t_2, \ldots, t_P$ are equal to $t_0$ as $K$ out of $P$ learners were updated at time $t_0$ and the remaining ($P - K$) of these $t_1, t_2, \ldots, t_P$ are $< t_0$. Let $Y_1, Y_2, \ldots, Y_P$ be the random variables denoting the computation time of the $P$ learners starting from time $t_0$. Thus,

$$Y_i = X_i - (t_0 - t_i) | X_i > (t_0 - t_i) \quad \forall i = 1, 2, \ldots, P \quad \text{(22)}$$

Now each of the $Y_i$s are independent and are stochastically dominated by the corresponding $X_i$ s.

$$\Pr(Y_i > u) \leq \Pr(X_i > u) \quad \forall i, j = 1, 2, \ldots, P \quad \text{(23)}$$

The expectation of the $K$-th statistic of $\{Y_1, Y_2, \ldots, Y_P\}$ is the runtime of the iteration. Let us denote $h_K(x_1, x_2, \ldots, x_P)$ as the $K$-th statistic of $P$ numbers $(x_1, x_2, \ldots, x_P)$. And let us denote $g_{K,s}(x)$ as the $K$-th statistic of $P$ numbers where $P - 1$ of them are given as $s_{1 \times (P-1)}$ and $x$ is the $P$-th number. Thus

$$g_{K,s}(x) = h_K(x, s(1), s(2), \ldots, s(P - 1)).$$

First observe that $g_{K,s}(x)$ is an increasing function of $x$ since given the other $P - 1$ values, the $K$-th order statistic will either stay the same or increase with $x$.

Now we use the property that if $Y_i$ is stochastically dominated by $X_i$, then for any increasing function $g(.)$, we have

$$E_{Y_i}[g(Y_i)] \leq E_{X_i}[g(X_i)].$$

This result is derived in [Kreps, 1990].

This implies that for a given $s$,

$$E_{Y_i}[g_{K,s}(Y_i)] \leq E_{X_i}[g_{K,s}(X_i)].$$

This leads to,

$$E_{Y_i}[g_{K,s}(Y_i)] \leq E_{X_i}[g_{K,s}(X_i)].$$

From this,

$$E[h_K(Y_1, Y_2, \ldots, Y_P)] = E_{Y_1}E_{Y_2}E_{Y_3}E_{Y_P}[h_K(Y_1, Y_2, \ldots, Y_P)] \leq E_{X_1}E_{X_2}E_{X_3}E_{X_P}[h_K(X_1, X_2, Y_3, \ldots, Y_P)] \quad \text{(24)}$$

This step proceeds inductively. Thus, similarly

$$E[h_K(X_1, X_2, \ldots, Y_P)] = E_{X_1}E_{X_2}E_{X_3}E_{X_P}[h_K(X_1, X_2, Y_3, \ldots, Y_P)] \leq E_{X_1}E_{X_2}E_{X_3}E_{X_P}[h_K(X_1, X_2, Y_3, \ldots, Y_P)] \quad \text{(26)}$$

Thus, finally combining, we have,

$$E[h_K(Y_1, Y_2, \ldots, Y_P)] \leq E[h_K(X_1, X_2, Y_3, \ldots, Y_P)] \leq E[h_K(X_1, X_2, X_3 \ldots X_P)] \quad \text{(27)}$$

**7.3.1 Special Case: Exponential Distributions**

For exponential distributions, the inequality in Lemma 5 holds with equality. This follows from the memoryless property of exponentials. Let us consider
the scenario of the proof of Lemma 5 where we similarly define \( Y_i = X_i - (t_0 - t_i) \mid X_i > (t_0 - t_i) \). From the memoryless property of exponentials [Sheldon, 2002], if \( X_i \sim \exp(\mu) \), then \( Y_i \sim \exp(\mu) \). Thus, the expectation of the \( K \)-th statistic of \( Y_i \)s can be easily derived as all the \( Y_i \)s are now i.i.d. with distribution \( \exp(\mu) \). Thus, the runtime for \( J \) iterations is given by,

\[
E[T] = J E[Y_{K:P}] = \frac{J}{\mu} \sum_{i=1}^{P-K+1} \frac{1}{i} \approx \frac{J}{\mu} \log \frac{P}{P-K}.
\]

7.3.2 Comparison of \( K \)-async and \( K \)-batch-async SGD

We compare the error-runtime trade-off of \( K \)-async with \( K \)-batch-async SGD in Figure 10 as follows.

Figure 10: Accuracy Runtime Trade-off on MNIST Dataset: Comparison of \( K \)-async with \( K \)-batch-async under exponential computation time with \( X_i \sim \exp(1) \). As derived theoretically, the \( K \)-batch-async has a sharper fall with time as compared to \( K \)-async even though the error attained is similar.

8 ASYNC-SGD ANALYSIS PROOFS

In this section, we provide a proof of the error convergence of asynchronous and \( K \)-async SGD.

8.1 Async-SGD with Fixed learning rate

First we prove a simplified version of Theorem 3 for the case \( K = 1 \). While this is actually a corollary of the more general Theorem 3, we prove this first for ease of understanding and simplicity. The proof of the more general Theorem 3 is then provided in Section 8.2.

Corollary 2. Suppose that the objective function \( F(w) \) is strongly convex with parameter \( c \) and the learning rate \( \eta \leq \frac{1}{2L(2c + e^c) + 1} \). Also assume that

\[
E[\|\nabla F(w_j) - \nabla F(w_{\tau(j)})\|^2] \leq \gamma E[\|\nabla F(w_j)\|^2]
\]

for some constant \( \gamma \leq 1 \). Then, the error after \( J \) iterations of Async SGD is given by,

\[
E[F(w_J)] - F^* \leq \eta L \alpha^2 + \frac{\eta L \alpha^2}{2 c^2} + (1 - \gamma \eta)^{J} (E[F(w_0)] - F^* - \eta L \alpha^2) \frac{1}{2 c^2}
\]

where \( \gamma = 1 - \gamma + \frac{\eta}{p_0} \) and \( p_0 \) is a non-negative lower bound on the conditional probability that \( \tau(j) = j \) given all the past delays and parameters.

To prove the result, we will use the following lemma.

Lemma 7. Let us denote \( v_j = g(w_{\tau(j)}, \xi_j) \), and assume that \( E_{\xi_j|w}[g(w, \xi_j)] = \nabla F(w) \). Then,

\[
E[\|\nabla F(w_j) - v_j\|^2] \leq E[\|v_j\|^2] - E[\|\nabla F(w_{\tau(j)})\|^2] + E[\|\nabla F(w_j) - \nabla F(w_{\tau(j)})\|^2].
\]

Proof of Lemma 7. Observe that,

\[
E[\|\nabla F(w_j) - v_j\|^2] = E[\|\nabla F(w_j) - \nabla F(w_{\tau(j)}) + \nabla F(w_{\tau(j)}) - v_j\|^2] = E[\|\nabla F(w_j) - \nabla F(w_{\tau(j)})\|^2] + E[\|\nabla F(w_j) - \nabla F(w_{\tau(j)})\|^2] - 2E[\nabla F(w_j) - \nabla F(w_{\tau(j)})] (28)
\]

The last line holds since the cross term is 0 as derived below.

\[
E[(\nabla F(w_j) - \nabla F(w_{\tau(j)}))^T (v_j - \nabla F(w_{\tau(j)}))] = E_{w_{\tau(j)}, w_j}[\nabla F(w_j) - \nabla F(w_{\tau(j)})]^T E_{\xi_j|w_{\tau(j)}, w_j}[v_j - \nabla F(w_{\tau(j)})] = E_{w_{\tau(j)}, w_j}[\nabla F(w_j) - \nabla F(w_{\tau(j)})]^T (E_{\xi_j|w_{\tau(j)}}[v_j] - \nabla F(w_{\tau(j)})) = 0
\]

Here again the last line follows from Assumption 2 in Section 2 which states that

\[
E_{\xi_j|w_{\tau(j)}}[v_j] = \nabla F(w_{\tau(j)}).
\]

Returning to (28), observe that the second term can be further decomposed as,

\[
E[\|v_j - \nabla F(w_{\tau(j)})\|^2] = E_{w_{\tau(j)}}[E_{\xi_j|w_{\tau(j)}}[\|v_j - \nabla F(w_{\tau(j)})\|^2]] = E_{w_{\tau(j)}}[E_{\xi_j|w_{\tau(j)}}[\|v_j\|^2]] - 2E_{w_{\tau(j)}}[E_{\xi_j|w_{\tau(j)}}[v_j^T \nabla F(w_{\tau(j)})]] + E_{w_{\tau(j)}}[E_{\xi_j|w_{\tau(j)}}[\|\nabla F(w_{\tau(j)})\|^2]] = E[\|v_j\|^2] - 2E[\|\nabla F(w_{\tau(j)})\|^2] + E[\|\nabla F(w_{\tau(j)})\|^2] = E[\|v_j\|^2] - E[\|\nabla F(w_{\tau(j)})\|^2].
\]

□
We also prove a $K$-learner version of this lemma to prove Theorem 3. Now we proceed to provide the proof of Corollary 2.

**Proof of Corollary 2.**

\[
F(w_{j+1}) \leq F(w_j) + (w_{j+1} - w_j)^T \nabla F(w_j) + \frac{L}{2}\|w_{j+1} - w_j\|^2
\]

Now, again bounding from (28),

\[
F(w_j) = (-\eta v_j)^T \nabla F(w_j) + \frac{L\eta^2}{2}\|v_j\|^2
\]

\[
F(w_j) - \frac{\eta}{2}\|\nabla F(w_j) - v_j\|^2 + \frac{L\eta^2}{2}\|v_j\|^2
\]

Here the last line follows from $2a^Tb = ||a||^2 + ||b||^2 - ||a - b||^2$. Taking expectation,

\[
E[F(w_{j+1})] \leq E[F(w_j)] - \frac{\eta}{2}E[||\nabla F(w_j)||^2]
- \frac{\eta}{2}E[||v_j||^2] + \frac{L\eta^2}{2}E[||v_j||^2]
\]

\[
= E[F(w_j)] - \frac{\eta}{2}\|\nabla F(w_j) - v_j\|^2 + \frac{L\eta^2}{2}\|v_j\|^2
\]

Using the above, we have

\[
E[F(w_{j+1})] \leq E[F(w_j)] - \frac{\eta}{2}\|\nabla F(w_j) - v_j\|^2 + \frac{L\eta^2}{2}\|v_j\|^2
\]

\[
E[F(w_{j+1})] \leq E[F(w_j)] - \frac{\eta}{2}\|\nabla F(w_j) - v_j\|^2 + \frac{L\eta^2}{2}\|v_j\|^2
\]

Here (a) follows from Lemma 7 that we just derived. Now, again bounding from (30), we have

\[
E[F(w_{j+1})] \leq E[F(w_j)] - \frac{\eta}{2}\|\nabla F(w_j) - v_j\|^2 + \frac{L\eta^2}{2}\|v_j\|^2
\]

\[
= E[F(w_j)] - \frac{\eta}{2}\|\nabla F(w_j) - v_j\|^2 + \frac{L\eta^2}{2}\|v_j\|^2
\]

Here (b) follows from the statement of the theorem that

\[
E[||\nabla F(w_j) - \nabla F(w_{\tau(j)})||^2] \leq \gamma E[||\nabla F(w_j)||^2]
\]

for some constant $\gamma \leq 1$. The next step (c) follows from Assumption 4 in Section 2 which lead to

\[
E[||v_j||^2] \leq \frac{\sigma^2}{m} + \left(\frac{M_G}{m} + 1\right) E[||\nabla F(w_{\tau(j)})||^2].
\]

Step (d) follows from choosing $\eta < \frac{1}{2L(m\sigma^2 + 1)}$ and finally (e) follows from Lemma 1.

Now one might recall that the function $F(w)$ was defined to be strongly convex with parameter $c$. Using the standard result of strong-convexity (6) in (32), we obtain the following result.

\[
E[F(w_{j+1})] - F^* \leq \frac{\eta^2L\sigma^2}{2m} + (1 - \eta c(1 - \gamma + \frac{P_0}{2}))(E[F(w_j)] - F^*)
\]

Let us denote $\gamma' = (1 - \gamma + \frac{P_0}{2})$. Then, using the above recursion, we thus have,

\[
E[F(w_j)] - F^* \leq \frac{\eta L\sigma^2}{2c\gamma'm} + (1 - \eta \gamma' c)(E[F(w_0)] - F^*) - \frac{\eta L\sigma^2}{2c\gamma' m}
\]

\[
\square
\]

### 8.1.1 Discussion on range of $p_0$

Let us denote the conditional probability of $\tau(j) = j$ given all the past delays and parameters as $p_0^{(j)}$. Now $p_0 \leq p_0^{(j)} \forall j$. Clearly the value of $p_0^{(j)}$ will differ for different distributions and accordingly the value of $p_0$ will differ. Here we include a brief discussion on the possible values of $p_0$ for different distributions. These also hold for $K$-async and $K$-batch-async SGD.

**Lemma 8 (Bounds of $p_0$).** Define $p_0 = \inf_j p_0^{(j)}$, i.e. the largest constant such that $p_0 \leq p_0^{(j)} \forall j$.

- For exponential computation times, $p_0^{(j)} = \frac{1}{j}$ for all $j$ and is thus invariant of $j$ and $p_0 = \frac{1}{j}$.
- For new-longer-than-used (See Definition 3) computation times, $p_0^{(j)} \leq \frac{1}{j}$ and thus $p_0 \leq \frac{1}{j}$.
- For new-shorter-than-used computation times, $p_0^{(j)} \geq \frac{1}{j}$ and thus $p_0 \geq \frac{1}{j}$.

**Proof of Lemma 8.** Let $t_0$ be the time when the $j$-th iteration occurs, and suppose that learner $i'$ pushed its gradient in the $j$-th iteration. Now similar to the
proof of Lemma 5, let us also assume that the learners last read their parameter values at time instants \(t_1, t_2, \ldots, t_p\) respectively where \(t'_j = t_0\) and the remaining \((P - 1)\) of these \(t_i\)'s are \(< t_0\). Let \(Y_1, Y_2, \ldots, Y_P\) be the random variables denoting the computation time of the \(P\) learners starting from time \(t_0\). Thus, \(Y_i = X_i - (t_0 - t_i)\). For exponents, from the memoryless property, all these \(Y_i\)'s become i.i.d. and thus from symmetry the probability of \(i'\) finishing before all the others is equal, i.e. \(\frac{1}{P}\). Thus, \(p^{(j)}_0 = p_0 = \frac{1}{P}\). For new-longer-than-used distributions, as we have discussed before all the \(Y_i\)'s with \(i \neq i'\) will be stochastically dominated by \(Y_i' = X_i'\). Thus, probability of is with \(i \neq i'\) finishing first is higher than \(i'\). Thus, \(p^{(j)}_0 \leq \frac{1}{P}\) and so is \(p_0\). Similarly, for new-shorter-than-used distributions, \(Y_i'\) is stochastically dominated by all the \(Y_i\)'s and thus probability of \(i'\) finishing first is more. So, \(p^{(j)}_0 \geq \frac{1}{P}\) and so is \(p_0\). 

\[ \text{Thus,} \]

\[ \mathbb{E}_{\xi_1, \ldots, \xi_K, \omega_{t_1}} \left[ \left\| \sum_{l=1}^{K} g(w_{l,t}, \xi_{l,t}) - F(w_{r(t)}) \right\|^2 \right] \]

\[ \leq \frac{K^2 \sigma^2}{m} + \left( \frac{MC}{m} + K \right) \left\| \sum_{l=1}^{K} \nabla F(w_{r(t)}) \right\|^2 \] (33)

\[ \text{Proof.} \]

First, let us consider the expectation of any cross term such that \(l \neq l'\). For the case of writing, let \(\Omega = \{w_{r(t_1)}, \ldots, w_{r(K)}\}\). Now observe the conditional expectation of the cross term as follows.

\[ \mathbb{E}_{\xi_1, \ldots, \xi_K, \omega_{t_1}} \left[(g(w_{l_1}, \xi_{l_1}))^T - \nabla F(w_{r(t_1)})) \right] \]

\[ \mathbb{E}_{\xi_1, \ldots, \xi_{l_1}, \omega_{t_1}} \left[(g(w_{l_1}, \xi_{l_1}))^T - \nabla F(w_{r(t_1)})) \right] \]

\[ \mathbb{E}_{\xi_1, \ldots, \xi_{l_1}, \omega_{t_1}} \left[(g(w_{l_1}, \xi_{l_1}))^T - \nabla F(w_{r(t_1)})) \right] \]

\[ = \mathbb{E}_{\xi_1, \ldots, \xi_{l_1}, \omega_{t_1}} \left[(g(w_{l_1}, \xi_{l_1}))^T - \nabla F(w_{r(t_1)})) \right] \]

\[ = \mathbb{E}_{\xi_1, \ldots, \xi_{l_1}, \omega_{t_1}} \left[(g(w_{l_1}, \xi_{l_1}))^T - \nabla F(w_{r(t_1)})) \right] \]

\[ = \mathbb{E}_{\xi_1, \ldots, \xi_{l_1}, \omega_{t_1}} \left[(g(w_{l_1}, \xi_{l_1}))^T - \nabla F(w_{r(t_1)})) \right] \]

Thus the cross terms are all 0. So the expression simplifies as,

\[ \mathbb{E}_{\xi_1, \ldots, \xi_K, \omega_{t_1}} \left[ \left\| \sum_{l=1}^{K} g(w_{l_1}, \xi_{l_1}) - F(w_{r(t_1)})) \right\|^2 \right] \]

\[ \leq \frac{K^2 \sigma^2}{m} + \left( \frac{MC}{m} + K \right) \left\| \sum_{l=1}^{K} \nabla F(w_{r(t_1)})) \right\|^2 \] (35)

Thus,

\[ \mathbb{E}_{\xi_1, \ldots, \xi_K, \omega_{t_1}} \left[ \left\| \sum_{l=1}^{K} g(w_{l_1}, \xi_{l_1}) - F(w_{r(t_1)})) \right\|^2 \right] \]

\[ = \mathbb{E}_{\xi_1, \ldots, \xi_K, \omega_{t_1}} \left[ \left\| \sum_{l=1}^{K} g(w_{l_2}, \xi_{l_2}) - F(w_{r(t_2)})) \right\|^2 \right] \]

\[ + \mathbb{E}_{\xi_1, \ldots, \xi_K, \omega_{t_1}} \left[ \left\| \sum_{l=1}^{K} F(w_{r(t_1)})) \right\|^2 \right] \]

\[ \leq \frac{K^2 \sigma^2}{m} + \left( \frac{MC}{m} + K \right) \left\| \sum_{l=1}^{K} \nabla F(w_{r(t_1)})) \right\|^2 \] (36)

Now we return to the proof of the theorem. 

\[ \text{Proof of Theorem 3.} \]

Let \(v_j = \frac{1}{K} \sum_{l=1}^{K} g(w_{l_1}, \xi_{l_1})\). Following steps similar to the Async-SGD proof, from Lipschitz continuity we have the following.

\[ F(w_{j+1}) \leq F(w_j) + (w_{j+1} - w_j)^T \nabla F(w_j) + \frac{L}{2} \left\| w_{j+1} - w_j \right\|^2 \]

\[ \leq F(w_j) - \frac{\eta}{K} \sum_{l=1}^{K} g(w_{l_1}, \xi_{l_1})^T \nabla F(w_j) + \frac{L \eta}{2} \left\| v_j \right\|^2 \]

\[ = F(w_j) - \frac{\eta}{K} \sum_{l=1}^{K} \left\| \nabla F(w_j) \right\|^2 - \frac{\eta}{2K} \sum_{l=1}^{K} \left\| g(w_{l_1}, \xi_{l_1}) \right\|^2 \]

\[ + \frac{\eta}{2K} \sum_{l=1}^{K} \left\| g(w_{l_1}, \xi_{l_1}) \right\|^2 - \frac{\eta}{2K} \sum_{l=1}^{K} \left\| g(w_{l_1}, \xi_{l_1}) \right\|^2 \]

\[ + \frac{L \eta^2}{2} \left\| v_j \right\|^2 \] (37)
Taking expectation,
\[ E[F(w_{j+1})] \leq E[F(w_j)] - \frac{\eta}{2} E[\|\nabla F(w_j)\|^2] \]
\[ - \frac{\eta}{2K} \sum_{l=1}^{K} E[\|g(w_{l,j}, \xi_{l,j})\|^2] \]
\[ + \frac{\eta}{2K} \sum_{l=1}^{K} E[\|\nabla F(w_j) - g(w_{l,j}, \xi_{l,j})\|^2] \]
\[ + \frac{L\eta^2}{2} E[\|v_j\|^2] \]

(a) follows from Lemma 7 and step (b) follows from the assumption that
\[ E[\|\nabla F(w_j) - \nabla F(w_{\tau(l,j)})\|^2] \leq \gamma E[\|\nabla F(w_j)\|^2] \]
for some constant \( \gamma \leq 1 \). The next step (c) follows from the Lemma 9 that bounds the variance of the sum of stochastic gradients. Step (d) follows from \( \eta < \frac{1}{2L(M_G + K)} \) and finally
\[ E[F(w_{j+1})] \leq E[F(w_j)] - \frac{\eta}{2} E[\|\nabla F(w_j)\|^2] \]
\[ - \frac{\eta}{2K} \sum_{l=1}^{K} E[\|g(w_{l,j}, \xi_{l,j})\|^2] \]
\[ + \frac{\eta}{2K} \sum_{l=1}^{K} E[\|\nabla F(w_j) - g(w_{l,j}, \xi_{l,j})\|^2] \]
\[ + \frac{L\eta^2}{2} E[\|v_j\|^2] \]

(e) follows from Lemma 1 in Section 3 that says
\[ E[\|\nabla F(w_{\tau(l,j)})\|^2] \geq \rho_0 E[\|\nabla F(w_j)\|^2] \]
for some non-negative constant \( \rho_0 \) which is a lower bound on the conditional probability that \( \tau(l,j) = j \) given all past delays and parameter values.

Finally, since \( F(w) \) is strongly convex, using the inequality \( 2\epsilon(F(w) - F^*) \leq E[\|\nabla F(w)\|^2] \) in (39), we finally obtain the desired result. \( \square \)

8.2.1 Extension to Non-Convex case

The analysis can be extended to provide weaker guarantees for non-convex objectives. Let \( \gamma' = 1 - \gamma + \frac{\rho_0}{2} \) for non-convex objectives, we have the following result.

Theorem 5. For non-convex objective function, we have the following ergodic convergence result given by:
\[ \frac{1}{J+1} \sum_{j=0}^{J} E[\|\nabla F(w_j)\|^2] \leq \frac{2(F(w_0) - F^*)}{(J+1)\eta\gamma'} + \frac{L\eta\sigma^2}{Km\gamma'} \]

where \( F^* = \min_w F(w) \).

Proof. Recall the recursion derived in the last proof in (39). After re-arrangement, we obtain the following:
\[ \frac{1}{J+1} \sum_{j=0}^{J} E[\|\nabla F(w_j)\|^2] \leq \frac{2(E[F(w_0)] - E[F(w_{j+1})])}{\eta\gamma'} + \frac{L\eta\sigma^2}{Km\gamma'} \]

Taking summation from \( j = 0 \) to \( j = J \), we get,
\[ \frac{1}{J+1} \sum_{j=0}^{J} E[\|\nabla F(w_j)\|^2] \]
\[ \leq \frac{2(F(w_0) - F^*)}{(J+1)\eta\gamma'} + \frac{L\eta\sigma^2}{Km\gamma'} \]

Here (a) follows since we assume \( w_0 \) to be known and also from \( E[F(w_j)] \geq F^* \). \( \square \)

8.3 Variable Learning Rate Schedule

We propose a new heuristic for learning rate schedule that is more stable than fixed learning rate for asynchronous SGD. Our learning rate schedule is \( \eta_j = \min \left\{ \frac{c}{\|w_j - w_{\tau(j)}\|^2}, \eta_{\text{max}} \right\} \), where \( \eta_{\text{max}} \) is a suitably large value of learning rate beyond which the convergence diverges. This heuristic is inspired from the assumption in Theorem 4 given by \( \eta_j E[\|w_j - w_{\tau(j)}\|^2] \leq C \). In this section, we derive the accuracy trade-off mentioned in Theorem 4 based on this assumption.
Proof of Theorem 4. Following steps similar to (29), we first obtain the following:

\[
F(w_{j+1}) \leq F(w_j) - \frac{\eta_j}{2} \left( \|\nabla F(w_j)\|^2 - \|\nabla F(w_{j+1})\|^2 \right) + \frac{\eta_j}{4} \left( \|v_j\|^2 - \|v_{j+1}\|^2 \right) + \frac{L\eta_j^2}{2} \|v_j\|^2 \quad (42)
\]

Now taking expectation, we obtain the following result.

\[
E[F(w_{j+1})] \leq E[F(w_j)] - \frac{\eta_j}{2} \left( \|\nabla F(w_j)\|^2 - \|\nabla F(w_{j+1})\|^2 \right) + \frac{\eta_j}{4} \left( \|v_j\|^2 - \|v_{j+1}\|^2 \right) + \frac{L\eta_j^2}{2} \|v_j\|^2 \quad (43)
\]

Here (a) follows from (30), (b) follows from (12), (c) follows from Assumption 4 and (d) follows as \( \eta_j \leq \frac{1}{2L(\frac{1}{m} + 1)} \). Let us define \( \Delta_j = \frac{CL^2}{2} + \frac{\eta_j^2 L^2}{2m} \). Thus, the recursion can be written as,

\[
E[F(w_{j+1})] \leq E[F(w_j)] - \frac{\eta_j}{2} \left( \|\nabla F(w_j)\|^2 - \|\nabla F(w_{j+1})\|^2 \right) + \frac{\eta_j}{4} \left( \|v_j\|^2 - \|v_{j+1}\|^2 \right) + \frac{L\eta_j^2}{2} \|v_j\|^2 + \Delta_j \quad (44)
\]

Here (e) follows from Lemma 1. If the loss function \( F(w) \) is strongly convex with parameter \( c \), then for all \( w \), we have \( 2c(F(w) - F^*) \leq \|\nabla F(w)\|^2 \). Using this result, we obtain

\[
E[F(w_{j+1})] - F^* \leq (1 - \eta_j (1 + \frac{p_0}{2})c)(E[F(w_j)] - F^*) + \Delta_j \\
\leq (1 - \eta_j (1 + \frac{p_0}{2})c)(1 - \eta_j - (1 + \frac{p_0}{2})c)(E[F(w_{j-1})] - F^*) + \Delta_j \\
+ (1 - \eta_j (1 + \frac{p_0}{2})c)\Delta_{j-1} + \Delta_j \\
\leq (1 - \rho_j)(1 - \rho_{j-1})... (1 - \rho_0)(E[F(w_0)] - F^*) + \Delta \\
\]

where \( \rho_j = \eta_j (1 + \frac{p_0}{2})c \) and \( \Delta_j = \Delta_j + (1 - \rho_j)\Delta_{j-1} + \cdots + (1 - \rho_j)(1 - \rho_{j-1})\cdots(1 - \rho_1)\Delta_0 \).

\[ \square \]

9 SIMULATION SETUP DETAILS

MNIST [LeCun, 1998]: For the simulations on MNIST dataset, we first convert the 28 \times 28 images into single vectors of length 784. We use a single layer of neurons followed by soft-max cross entropy with logits loss function. Thus effectively the parameters consist of a weight matrix \( W \) of size 784 \times 10 and a bias vector \( b \) of size 1 \times 10. We use a regularizer of value 0.01, mini-batch size \( m = 1 \), and learning rate \( \eta = 0.01 \). For implementation we used Tensorflow with Python3. Thus, the model is as follows:

\[
X=tf.placeholder(tf.float32,[None,784]) \\
Y=tf.placeholder(tf.float32,[None,10]) \\
W=tf.Variable(tf.random_normal(shape=[784,10], stddev=0.01), name=\"weights\") \\
b=tf.Variable(tf.random_normal(shape=[1,10], stddev=0.01), name=\"bias\") \\
logits=tf.matmul(X,W) + b \\
tf.nn.softmax_cross_entropy_with_logits(logits=logits,labels=Y) + lamda*tf.reduce_mean(tf.square(tf.norm(W))) \\
loss=tf.reduce_mean(entropy)
\]

For the run-time simulations, we generate random variables from the respective distributions in python to represent the computation times.

CIFAR10 [Krizhevsky and Hinton, 2009]: For the CIFAR10 simulations, similar to MNIST, we convert the images into vectors of length 1024. We combine the three colour variants in the ratio [0.2989, 0.5870, 0.114] to generate a single vector of length 1024 for every image. We use a single layer of neurons again followed by soft-max cross entropy with logits in tensorflow. Thus, the parameters consist of a weight matrix \( W \) of size 1024 \times 10 and a bias vector \( b \) of size 1 \times 10. We use a mini-batch size of 250, regularizer of 0.05.

We use a similar model as follows:

\[
X=tf.placeholder(tf.float32,[None,1024]) \\
Y=tf.placeholder(tf.float32,[None,10]) \\
W=tf.Variable(tf.random_normal(shape=[1024,10], stddev=0.01), name=\"weights\") \\
b=tf.Variable(tf.random_normal(shape=[1,10], stddev=0.01), name=\"bias\") \\
logits=tf.matmul(X,W) + b \\
tf.nn.softmax_cross_entropy_with_logits(logits=logits,labels=Y) + lamda*tf.reduce_mean(tf.square(tf.norm(W))) \\
loss=tf.reduce_mean(entropy) \\
\]
Slow and Stale Gradients Can Win the Race: Error-Runtime Trade-offs in Distributed SGD

Figure 11: Error-Iterations tradeoff on MNIST dataset: Simulation of $K$-sync SGD for different values of $K$. Observe that accuracy improves with increasing $K$ which means increasing effective batch size ($\eta = 0.05$).

Figure 12: Error-Runtime tradeoff on MNIST dataset: Simulation of $K$-sync SGD for different values of $K$ ($\eta = 0.05$).

$\text{loss} = \text{tf.reduce_mean(entropy)}$

The computation time as each learner is generated from exponential distribution.

10 CHOICE OF HYPERPARAMETERS

Our analysis techniques can also inform the choice of hyperparameters for synchronous and $K$-sync SGD.

10.1 Varying $K$ in $K$-sync

We first perform some simulations of $K$-sync SGD applied on the MNIST dataset. For the simulation set-up, we consider 8 parallel learners with fixed mini-batch size $m = 1$ and fixed learning rate 0.05. The number of learners to wait for in $K$-sync, i.e. $K$ is varied and the error-runtime trade-off is observed. The runtimes are generated from a shifted exponential distribution given by $X_i \sim m + \exp \mu$, that depends on the mini-batch size. Intuitively, this distribution makes sense since to compute one mini-batch, a learner would at least need a time $m$ (Work Complexity). However, due to delays, it has the additional exponential tail. The error-runtime trade-offs are observed in Figure 13 and Figure 14.

(See Figure 12) instead of the number of iterations, observe that increasing $K$ naively does not always lead to a better trade-off. As $K$ increases, the central PS has to wait for more learners to finish at every iteration, thus suffering from increased straggler effect. The best error-runtime trade-off is obtained at an intermediate $K = 4$. Thus, the current analysis informs the optimal choice of $K$ to achieve a good error-runtime trade-off.

10.2 Varying mini-batch $m$

We consider the training of Alexnet on ImageNet dataset [Krizhevsky et al., 2012] using $P = 4$ learners. For this simulation, we perform fully synchronous SGD, i.e. $K$-sync with $K = P = 4$. We fix the learning rate and vary the mini-batch used for training. The runtimes are generated from a shifted exponential distribution given by $X_i \sim m + \exp \mu$, that depends on the mini-batch size. Intuitively, this distribution makes sense since to compute one mini-batch, a learner would at least need a time $m$ (Work Complexity). However, due to delays, it has the additional exponential tail.

The error-runtime trade-offs are observed in Figure 13 and Figure 14.

Figure 13: Error-Iterations tradeoff on IMAGENET dataset: Simulation of fully synchronous SGD ($K = P = 4$) for different values of mini-batch $m$. Observe that accuracy improves with increasing $m$ which means increasing effective batch size.

Figure 14: Error-Runtime tradeoff on IMAGENET dataset: Same simulation of fully synchronous SGD ($K = P = 4$) for different values of mini-batch $m$ plotted against time. Observe that higher $m$ does not necessarily mean the best trade-off with runtime as higher mini-batch also has longer time.
Again, observe that the plot of error with the number of iterations improves with the mini-batch size, as also expected from theory. However, increasing the mini-batch also changes the runtime distribution. Thus, when we plot the same error against runtime, we again observe that increasing the mini-batch size naively does not necessarily lead to the best trade-off. Instead, the best error-runtime trade-off is observed with an intermediate mini-batch value of 1024. Thus, our analysis informs the choice of the optimal mini-batch.