arXiv:1412.6651v8 [cs.LG] 25 Oct 2015

Deep learning with Elastic Averaging SGD

Sixin Zhang
Courant Institute, NYU
zsx@cims.nyu.edu

Anna Choromanska
Courant Institute, NYU
achoroma@cims.nyu.edu

Yann LeCun
Center for Data Science, NYU & Facebook AI Research
yann@cims.nyu.edu

Abstract

在此背景下，提出了一种新的算法，即并行进程之间的工作通信和协调是基于弹性力，该弹性力将计算的参数与参数服务器(master)存储的中心变量联系起来。

We study the problem of stochastic optimization for deep learning in the parallel computing environment under communication constraints. A new algorithm is proposed in this setting where the communication and coordination of work among concurrent processes (local workers), is based on an elastic force which links the parameters they compute with a center variable stored by the parameter server (master). The algorithm enables the local workers to perform more exploration, i.e. the algorithm allows the local variables to fluctuate further from the center variable by reducing the amount of communication between local workers and the master. We empirically demonstrate that in the deep learning setting, due to the existence of many local optima, allowing more exploration can lead to the improved performance. We propose synchronous and asynchronous variants of the new algorithm. We provide the stability analysis of the asynchronous variant in the round-robin scheme and compare it with the more common parallelized method *ADMM*. We show that the stability of *EASGD* is guaranteed when a simple stability condition is satisfied, which is not the case for *ADMM*. We additionally propose the momentum-based version of our algorithm that can be applied in both synchronous and asynchronous settings. Asynchronous variant of the algorithm is applied to train convolutional neural networks for image classification on the *CIFAR* and *ImageNet* datasets. Experiments demonstrate that the new algorithm accelerates the training of deep architectures compared to *DOWNPOUR* and other common baseline approaches and furthermore is very communication efficient.

该算法通过减少本地工作者和主变量之间的通信量，允许局部变量进一步波动。

1 Introduction

挑战

One of the most challenging problems in large-scale machine learning is how to parallelize the training of large models that use a form of stochastic gradient descent (*SGD*) [1]. There have been attempts to parallelize *SGD*-based training for large-scale deep learning models on large number of CPUs, including the Google's Distbelief system [2]. But practical image recognition systems consist of large-scale convolutional neural networks trained on few GPU cards sitting in a single computer [3, 4]. The main challenge is to devise parallel *SGD* algorithms to train large-scale deep learning models that yield a significant speedup when run on multiple GPU cards.

动力来自二次惩罚法

In this paper we introduce the *Elastic Averaging SGD* method (*EASGD*) and its variants. *EASGD* is motivated by quadratic penalty method [5], but is re-interpreted as a parallelized extension of the averaging *SGD* algorithm [6]. The basic idea is to let each worker maintain its own local parameter, and the communication and coordination of work among the local workers is based on an elastic force which links the parameters they compute with a center variable stored by the master. The center variable is updated as a moving average where the average is taken in time and also in space over the parameters computed by local workers. The main contribution of this paper is a new algorithm that provides fast convergent minimization while outperforming *DOWNPOUR* method [2] and other

贡献

baseline approaches in practice. Simultaneously it reduces the communication overhead between the master and the local workers while at the same time it maintains high-quality performance measured by the test error. The new algorithm applies to deep learning settings such as parallelized training of convolutional neural networks.

The article is organized as follows. Section 2 explains the problem setting, Section 3 presents the synchronous *EASGD* algorithm and its asynchronous and momentum-based variants, Section 4 provides stability analysis of *EASGD* and *ADMM* in the round-robin scheme, Section 5 shows experimental results and Section 6 concludes. The Supplement contains additional material including additional theoretical analysis.

2 Problem setting

Consider minimizing a function $F(x)$ in a parallel computing environment [7] with $p \in \mathbb{N}$ workers and a master. In this paper we focus on the stochastic optimization problem of the following form

公式符号看不懂

$$\min_x F(x) := \mathbb{E}[f(x, \xi)], \tag{1}$$

where x is the model parameter to be estimated and ξ is a random variable that follows the probability distribution $\mathbb{P}$ over Ω such that $F(x) = \int_\Omega f(x, \xi)\mathbb{P}(d\xi)$. The optimization problem in Equation 1 can be reformulated as follows

$$\min_{x^1,\ldots,x^p,\tilde{x}} \sum_{i=1}^{p} \mathbb{E}[f(x^i, \xi^i)] + \frac{\rho}{2}\|x^i - \tilde{x}\|^2, \tag{2}$$

where each ξ^i follows the same distribution $\mathbb{P}$ (thus we assume each worker can sample the entire dataset). In the paper we refer to x^i's as local variables and we refer to $\tilde{x}$ as a center variable. The problem of the equivalence of these two objectives is studied in the literature and is known as the *augmentability* or the *global variable consensus* problem [8, 9]. The quadratic penalty term ρ in Equation 2 is expected to ensure that local workers will not fall into different attractors that are far away from the center variable. This paper focuses on the problem of reducing the parameter communication overhead between the master and local workers [10, 2, 11, 12, 13]. The problem of data communication when the data is distributed among the workers [7, 14] is a more general problem and is not addressed in this work. We however emphasize that our problem setting is still highly non-trivial under the communication constraints due to the existence of many local optima [15].

3 EASGD update rule

The *EASGD* updates captured in resp. Equation 3 and 4 are obtained by taking the gradient descent step on the objective in Equation 2 with respect to resp. variable x^i and $\tilde{x}$,

学习率×梯度

$$x^i_{t+1} = x^i_t - \eta(g^i_t(x^i_t) + \rho(x^i_t - \tilde{x}_t)) \tag{3}$$

$$\tilde{x}_{t+1} = \tilde{x}_t + \eta \sum_{i=1}^{p} \rho(x^i_t - \tilde{x}_t), \tag{4}$$

where $g^i_t(x^i_t)$ denotes the stochastic gradient of F with respect to x^i evaluated at iteration t, x^i_t and $\tilde{x}_t$ denote respectively the value of variables x^i and $\tilde{x}$ at iteration t, and η is the learning rate.

The update rule for the center variable $\tilde{x}$ takes the form of moving average where the average is taken over both space and time. Denote $\alpha = \eta\rho$ and $\beta = p\alpha$, then Equation 3 and 4 become

$$x^i_{t+1} = x^i_t - \eta g^i_t(x^i_t) - \alpha(x^i_t - \tilde{x}_t) \tag{5}$$

$$\tilde{x}_{t+1} = (1-\beta)\tilde{x}_t + \beta\left(\frac{1}{p}\sum_{i=1}^{p} x^i_t\right). \tag{6}$$

Note that choosing $\beta = p\alpha$ leads to an elastic symmetry in the update rule, i.e. there exists an symmetric force equal to $\alpha(x^i_t - \tilde{x}_t)$ between the update of each x^i and $\tilde{x}$. It has a crucial influence on the algorithm's stability as will be explained in Section 4. Also in order to minimize the staleness [16] of the difference $x^i_t - \tilde{x}_t$ between the center and the local variable, the update for the master in Equation 4 involves x^i_t instead of x^i_{t+1}.

Note also that $\alpha = \eta\rho$, where the magnitude of ρ represents the amount of exploration we allow in the model. In particular, small ρ allows for more exploration as it allows x^i's to fluctuate further from the center $\tilde{x}$. The distinctive idea of *EASGD* is to allow the local workers to perform more exploration (small ρ) and the master to perform exploitation. This approach differs from other settings explored in the literature [2, 17, 18, 19, 20, 21, 22, 23], and focus on how fast the center variable converges. In this paper we show the merits of our approach in the deep learning setting.

3.1 Asynchronous EASGD

We discussed the synchronous update of *EASGD* algorithm in the previous section. In this section we propose its asynchronous variant. The local workers are still responsible for updating the local variables x^i's, whereas the master is updating the center variable $\tilde{x}$. Each worker maintains its own clock t^i, which starts from 0 and is incremented by 1 after each stochastic gradient update of x^i as shown in Algorithm 1. The master performs an update whenever the local workers finished τ steps of their gradient updates, where we refer to τ as the *communication period*. As can be seen in Algorithm 1, whenever τ divides the local clock of the ith worker, the ith worker communicates with the master and requests the current value of the center variable $\tilde{x}$. The worker then waits until the master sends back the requested parameter value, and computes the elastic difference $\alpha(x - \tilde{x})$ (this entire procedure is captured in step a) in Algorithm 1). The elastic difference is then sent back to the master (step b) in Algorithm 1) who then updates $\tilde{x}$.

The communication period τ controls the frequency of the communication between every local worker and the master, and thus the trade-off between exploration and exploitation.

Algorithm 1: Asynchronous EASGD: Processing by worker i and the master

Input: learning rate η, moving rate α, communication period $\tau \in \mathbb{N}$
Initialize: $\tilde{x}$ is initialized randomly, $x^i = \tilde{x}$, $t^i = 0$

Repeat
  $x \leftarrow x^i$
  if (τ divides t^i) **then**
    a) $x^i \leftarrow x^i - \alpha(x - \tilde{x})$
    b) $\tilde{x} \leftarrow \tilde{x} + \alpha(x - \tilde{x})$
  end
  $x^i \leftarrow x^i - \eta g^i_{t^i}(x)$
  $t^i \leftarrow t^i + 1$
Until forever

Algorithm 2: Asynchronous EAMSGD: Processing by worker i and the master

Input: learning rate η, moving rate α, communication period $\tau \in \mathbb{N}$, momentum term δ
Initialize: $\tilde{x}$ is initialized randomly, $x^i = \tilde{x}$, $v^i = 0$, $t^i = 0$

Repeat
  $x \leftarrow x^i$
  if (τ divides t^i) **then**
    a) $x^i \leftarrow x^i - \alpha(x - \tilde{x})$
    b) $\tilde{x} \leftarrow \tilde{x} + \alpha(x - \tilde{x})$
  end
  $v^i \leftarrow \delta v^i - \eta g^i_{t^i}(x + \delta v^i)$
  $x^i \leftarrow x^i + v^i$
  $t^i \leftarrow t^i + 1$
Until forever

3.2 Momentum EASGD

The momentum EASGD (*EAMSGD*) is a variant of our Algorithm 1 and is captured in Algorithm 2. It is based on the Nesterov's momentum scheme [24, 25, 26], where the update of the local worker of the form captured in Equation 3 is replaced by the following update

$$\begin{aligned} v^i_{t+1} &= \delta v^i_t - \eta g^i_t(x^i_t + \delta v^i_t) \\ x^i_{t+1} &= x^i_t + v^i_{t+1} - \eta\rho(x^i_t - \tilde{x}_t), \end{aligned} \tag{7}$$

where δ is the momentum term. Note that when $\delta = 0$ we recover the original *EASGD* algorithm.

As we are interested in reducing the communication overhead in the parallel computing environment where the parameter vector is very large, we will be exploring in the experimental section the asynchronous *EASGD* algorithm and its momentum-based variant in the relatively large τ regime (less frequent communication).

4 Stability analysis of EASGD and ADMM in the round-robin scheme

In this section we study the stability of the asynchronous *EASGD* and *ADMM* methods in the round-robin scheme [20]. We first state the updates of both algorithms in this setting, and then we study

their stability. We will show that in the one-dimensional quadratic case, *ADMM* algorithm can exhibit chaotic behavior, leading to exponential divergence. The analytic condition for the *ADMM* algorithm to be stable is still unknown, while for the *EASGD* algorithm it is very simple[1].

The analysis of the synchronous *EASGD* algorithm, including its convergence rate, and its averaging property, in the quadratic and strongly convex case, is deferred to the Supplement.

In our setting, the *ADMM* method [9, 27, 28] involves solving the following minimax problem[2],

$$\max_{\lambda^1,\ldots,\lambda^p} \min_{x^1,\ldots,x^p,\tilde{x}} \sum_{i=1}^{p} F(x^i) - \lambda^i(x^i - \tilde{x}) + \frac{\rho}{2}\|x^i - \tilde{x}\|^2, \tag{8}$$

where λ^i's are the Lagrangian multipliers. The resulting updates of the *ADMM* algorithm in the round-robin scheme are given next. Let $t \geq 0$ be a global clock. At each t, we linearize the function $F(x^i)$ with $F(x^i_t) + \left\langle \nabla F(x^i_t), x^i - x^i_t \right\rangle + \frac{1}{2\eta}\left\|x^i - x^i_t\right\|^2$ as in [28]. The updates become

$$\lambda^i_{t+1} = \begin{cases} \lambda^i_t - (x^i_t - \tilde{x}_t) & \text{if} \quad \text{mod } (t,p) = i-1; \\ \lambda^i_t & \text{if} \quad \text{mod } (t,p) \neq i-1. \end{cases} \tag{9}$$

$$x^i_{t+1} = \begin{cases} \frac{x^i_t - \eta\nabla F(x^i_t) + \eta\rho(\lambda^i_{t+1} + \tilde{x}_t)}{1+\eta\rho} & \text{if} \quad \text{mod } (t,p) = i-1; \\ x^i_t & \text{if} \quad \text{mod } (t,p) \neq i-1. \end{cases} \tag{10}$$

$$\tilde{x}_{t+1} = \frac{1}{p}\sum_{i=1}^{p}(x^i_{t+1} - \lambda^i_{t+1}). \tag{11}$$

Each local variable x^i is periodically updated (with period p). First, the Lagrangian multiplier λ^i is updated with the dual ascent update as in Equation 9. It is followed by the gradient descent update of the local variable as given in Equation 10. Then the center variable $\tilde{x}$ is updated with the most recent values of all the local variables and Lagrangian multipliers as in Equation 11. Note that since the step size for the dual ascent update is chosen to be ρ by convention [9, 27, 28], we have re-parametrized the Lagrangian multiplier to be $\lambda^i_t \leftarrow \lambda^i_t/\rho$ in the above updates.

The *EASGD* algorithm in the round-robin scheme is defined similarly and is given below

$$x^i_{t+1} = \begin{cases} x^i_t - \eta\nabla F(x^i_t) - \alpha(x^i_t - \tilde{x}_t) & \text{if} \quad \text{mod } (t,p) = i-1; \\ x^i_t & \text{if} \quad \text{mod } (t,p) \neq i-1. \end{cases} \tag{12}$$

$$\tilde{x}_{t+1} = \tilde{x}_t + \sum_{i:\ \text{mod } (t,p)=i-1} \alpha(x^i_t - \tilde{x}_t). \tag{13}$$

At time t, only the i-th local worker (whose index $i-1$ equals t modulo p) is activated, and performs the update in Equations 12 which is followed by the master update given in Equation 13.

We will now focus on the one-dimensional quadratic case without noise, i.e. $F(x) = \frac{x^2}{2}, x \in \mathbb{R}$.

For the *ADMM* algorithm, let the state of the (dynamical) system at time t be $s_t = (\lambda^1_t, x^1_t, \ldots, \lambda^p_t, x^p_t, \tilde{x}_t) \in \mathbb{R}^{2p+1}$. The local worker i's updates in Equations 9, 10, and 11 are composed of three linear maps which can be written as $s_{t+1} = (F^i_3 \circ F^i_2 \circ F^i_1)(s_t)$. For simplicity, we will only write them out below for the case when $i = 1$ and $p = 2$:

$$F^1_1 = \begin{pmatrix} 1 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, F^1_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{\eta\rho}{1+\eta\rho} & \frac{1-\eta}{1+\eta\rho} & 0 & 0 & \frac{\eta\rho}{1+\eta\rho} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, F^1_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{1}{p} & \frac{1}{p} & -\frac{1}{p} & \frac{1}{p} & 0 \end{pmatrix}.$$

For each of the p linear maps, it's possible to find a simple condition such that each map, where the ith map has the form $F^i_3 \circ F^i_2 \circ F^i_1$, is stable (the absolute value of the eigenvalues of the map are

[1] This condition resembles the stability condition for the synchronous *EASGD* algorithm (Condition 17 for $p = 1$) in the analysis in the Supplement.

[2] The convergence analysis in [27] is based on the assumption that "At any master iteration, updates from the workers have the same probability of arriving at the master.", which is not satisfied in the round-robin scheme.

smaller or equal to one). However, when these non-symmetric maps are composed one after another as follows $\mathcal{F} = F_3^p \circ F_2^p \circ F_1^p \circ \ldots \circ F_3^1 \circ F_2^1 \circ F_1^1$, the resulting map $\mathcal{F}$ can become unstable! (more precisely, some eigenvalues of the map can sit outside the unit circle in the complex plane).

We now present the numerical conditions for which the *ADMM* algorithm becomes unstable in the round-robin scheme for $p = 3$ and $p = 8$, by computing the largest absolute eigenvalue of the map $\mathcal{F}$. Figure 1 summarizes the obtained result.

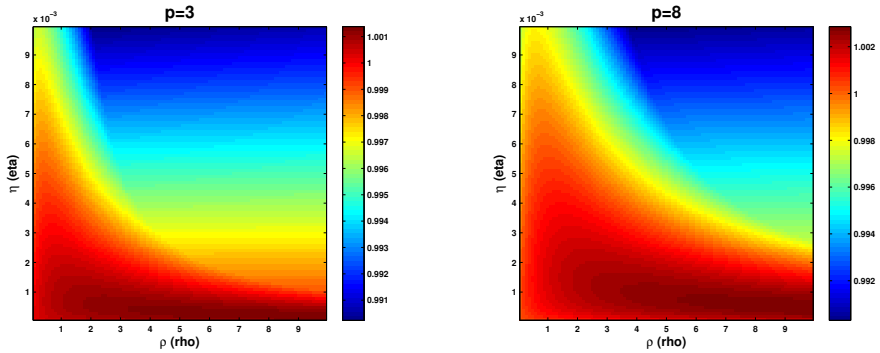

Figure 1: The largest absolute eigenvalue of the linear map $\mathcal{F} = F_3^p \circ F_2^p \circ F_1^p \circ \ldots \circ F_3^1 \circ F_2^1 \circ F_1^1$ as a function of $\eta \in (0, 10^{-2})$ and $\rho \in (0, 10)$ when $p = 3$ and $p = 8$. To simulate the chaotic behavior of the *ADMM* algorithm, one may pick $\eta = 0.001$ and $\rho = 2.5$ and initialize the state s_0 either randomly or with $\lambda_0^i = 0, x_0^i = \tilde{x}_0 = 1000, \forall i$. *Figure should be read in color.*

On the other hand, the *EASGD* algorithm involves composing only symmetric linear maps due to the elasticity. Let the state of the (dynamical) system at time t be $s_t = (x_t^1, \ldots, x_t^p, \tilde{x}_t) \in \mathbb{R}^{p+1}$. The activated local worker i's update in Equation 12 and the master update in Equation 13 can be written as $s_{t+1} = F^i(s_t)$. In case of $p = 2$, the map F^1 and F^2 are defined as follows

$$F^1 = \begin{pmatrix} 1-\eta-\alpha & 0 & \alpha \\ 0 & 1 & 0 \\ \alpha & 0 & 1-\alpha \end{pmatrix}, F^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1-\eta-\alpha & \alpha \\ 0 & \alpha & 1-\alpha \end{pmatrix}$$

For the composite map $F^p \circ \ldots \circ F^1$ to be stable, the condition that needs to be satisfied is actually the same for each i, and is furthermore independent of p (since each linear map F^i is symmetric). It essentially involves the stability of the 2×2 matrix $\begin{pmatrix} 1-\eta-\alpha & \alpha \\ \alpha & 1-\alpha \end{pmatrix}$, whose two (real) eigenvalues λ satisfy $(1-\eta-\alpha-\lambda)(1-\alpha-\lambda) = \alpha^2$. The resulting stability condition ($|\lambda| \leq 1$) is simple and given as $0 \leq \eta \leq 2, 0 \leq \alpha \leq \frac{4-2\eta}{4-\eta}$.

5 Experiments

In this section we compare the performance of *EASGD* and *EAMSGD* with the parallel method *DOWNPOUR* and the sequential method *SGD*, as well as their averaging and momentum variants.

All the parallel comparator methods are listed below[3]:

- *DOWNPOUR* [2], the pseudo-code of the implementation of *DOWNPOUR* used in this paper is enclosed in the Supplement.
- *Momentum DOWNPOUR* (*MDOWNPOUR*), where the Nesterov's momentum scheme is applied to the master's update (note it is unclear how to apply it to the local workers or for the case when $\tau > 1$). The pseudo-code is in the Supplement.
- A method that we call *ADOWNPOUR*, where we compute the average over time of the center variable $\tilde{x}$ as follows: $z_{t+1} = (1-\alpha_{t+1})z_t + \alpha_{t+1}\tilde{x}_t$, and $\alpha_{t+1} = \frac{1}{t+1}$ is a moving rate, and $z_0 = \tilde{x}_0$. t denotes the master clock, which is initialized to 0 and incremented every time the center variable $\tilde{x}$ is updated.
- A method that we call *MVADOWNPOUR*, where we compute the moving average of the center variable $\tilde{x}$ as follows: $z_{t+1} = (1-\alpha)z_t + \alpha\tilde{x}_t$, and the moving rate α was chosen to be constant, and $z_0 = \tilde{x}_0$. t denotes the master clock and is defined in the same way as for the *ADOWNPOUR* method.

[3]We have compared asynchronous *ADMM* [27] with *EASGD* in our setting as well, the performance is nearly the same. However, *ADMM*'s momentum variant is not as stable for large communication periods.

All the sequential comparator methods ($p = 1$) are listed below:

- *SGD* [1] with constant learning rate η.
- *Momentum SGD* (*MSGD*) [26] with constant momentum δ.
- *ASGD* [6] with moving rate $\alpha_{t+1} = \frac{1}{t+1}$.
- *MVASGD* [6] with moving rate α set to a constant.

We perform experiments in a deep learning setting on two benchmark datasets: CIFAR-10 (we refer to it as *CIFAR*) [4] and ImageNet ILSVRC 2013 (we refer to it as *ImageNet*) [5]. We focus on the image classification task with deep convolutional neural networks. We next explain the experimental setup. The details of the data preprocessing and prefetching are deferred to the Supplement.

5.1 Experimental setup

For all our experiments we use a GPU-cluster interconnected with InfiniBand. Each node has 4 Titan GPU processors where each local worker corresponds to one GPU processor. The center variable of the master is stored and updated on the centralized parameter server [2][6].

To describe the architecture of the convolutional neural network, we will first introduce a notation. Let (c, y) denotes the size of the input image to each layer, where c is the number of color channels and y is both the horizontal and the vertical dimension of the input. Let C denotes the fully-connected convolutional operator and let P denotes the max pooling operator, D denotes the linear operator with dropout rate equal to 0.5 and S denotes the linear operator with softmax output non-linearity. We use the cross-entropy loss and all inner layers use rectified linear units. For the *ImageNet* experiment we use the similar approach to [4] with the following 11-layer convolutional neural network (3,221)C(96,108)P(96,36)C(256,32)P(256,16)C(384,14)C(384,13)C(256,12)P(256,6)D(4096,1)D(4096,1)S(1000,1). For the *CIFAR* experiment we use the similar approach to [29] with the following 7-layer convolutional neural network (3,28)C(64,24)P(64,12)C(128,8)P(128,4)C(64,2)D(256,1)S(10,1).

In our experiments all the methods we run use the same initial parameter chosen randomly, except that we set all the biases to zero for *CIFAR* case and to 0.1 for *ImageNet* case. This parameter is used to initialize the master and all the local workers[7]. We add l_2-regularization $\frac{\lambda}{2}\|x\|^2$ to the loss function $F(x)$. For *ImageNet* we use $\lambda = 10^{-5}$ and for *CIFAR* we use $\lambda = 10^{-4}$. We also compute the stochastic gradient using mini-batches of sample size 128.

5.2 Experimental results

For all experiments in this section we use *EASGD* with $\beta = 0.9$[8] , for all momentum-based methods we set the momentum term $\delta = 0.99$ and finally for *MVADOWNPOUR* we set the moving rate to $\alpha = 0.001$. We start with the experiment on *CIFAR* dataset with $p = 4$ local workers running on a single computing node. For all the methods, we examined the communication periods from the following set $\tau = \{1, 4, 16, 64\}$. For comparison we also report the performance of *MSGD* which outperformed *SGD*, *ASGD* and *MVASGD* as shown in Figure 6 in the Supplement. For each method we examined a wide range of learning rates (the learning rates explored in all experiments are summarized in Table 1, 2, 3 in the Supplement). The *CIFAR* experiment was run 3 times independently from the same initialization and for each method we report its best performance measured by the smallest achievable test error. From the results in Figure 2, we conclude that all *DOWNPOUR*-based methods achieve their best performance (test error) for small τ ($\tau \in \{1, 4\}$), and become highly unstable for $\tau \in \{16, 64\}$. While *EAMSGD* significantly outperforms comparator methods for all values of τ by having faster convergence. It also finds better-quality solution measured by the test error and this advantage becomes more significant for $\tau \in \{16, 64\}$. Note that the tendency to achieve better test performance with larger τ is also characteristic for the *EASGD* algorithm.

[4]Downloaded from `http://www.cs.toronto.edu/~kriz/cifar.html`.

[5]Downloaded from `http://image-net.org/challenges/LSVRC/2013`.

[6]Our implementation is available at `https://github.com/sixin-zh/mpiT`.

[7]On the contrary, initializing the local workers and the master with different random seeds 'traps' the algorithm in the symmetry breaking phase.

[8]Intuitively the 'effective β' is $\beta/\tau = p\alpha = p\eta\rho$ (thus $\rho = \frac{\beta}{\tau p \eta}$) in the asynchronous setting.

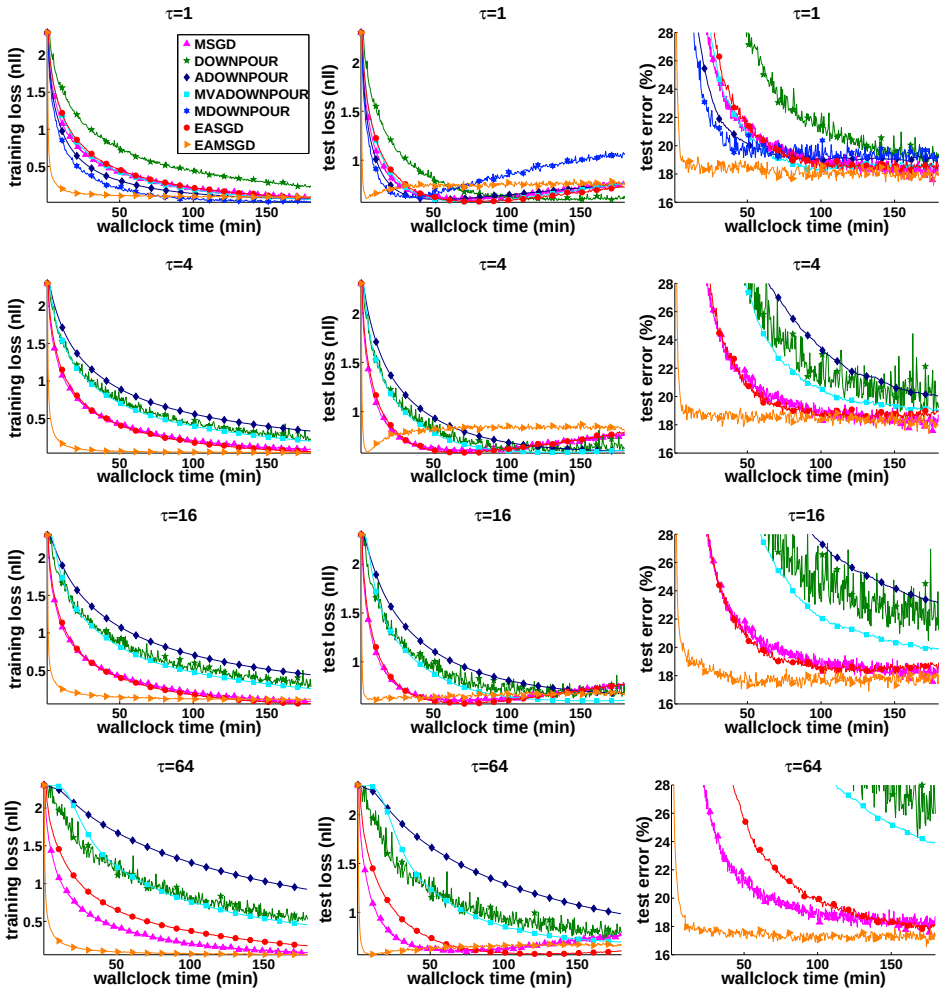

Figure 2: Training and test loss and the test error for the center variable versus a wallclock time for different communication periods τ on *CIFAR* dataset with the 7-layer convolutional neural network.

We next explore different number of local workers p from the set $p = \{4, 8, 16\}$ for the *CIFAR* experiment, and $p = \{4, 8\}$ for the *ImageNet* experiment[9]. For the *ImageNet* experiment we report the results of one run with the best setting we have found. *EASGD* and *EAMSGD* were run with $\tau = 10$ whereas *DOWNPOUR* and *MDOWNPOUR* were run with $\tau = 1$. The results are in Figure 3 and 4. For the *CIFAR* experiment, it's noticeable that the lowest achievable test error by either *EASGD* or *EAMSGD* decreases with larger p. This can potentially be explained by the fact that larger p allows for more exploration of the parameter space. In the Supplement, we discuss further the trade-off between exploration and exploitation as a function of the learning rate (section 9.5) and the communication period (section 9.6). Finally, the results obtained for the *ImageNet* experiment also shows the advantage of *EAMSGD* over the competitor methods.

6 Conclusion

In this paper we describe a new algorithm called *EASGD* and its variants for training deep neural networks in the stochastic setting when the computations are parallelized over multiple GPUs. Experiments demonstrate that this new algorithm quickly achieves improvement in test error compared to more common baseline approaches such as *DOWNPOUR* and its variants. We show that our approach is very stable and plausible under communication constraints. We provide the stability analysis of the asynchronous *EASGD* in the round-robin scheme, and show the theoretical advantage of the method over *ADMM*. The different behavior of the *EASGD* algorithm from its momentum-based variant *EAMSGD* is intriguing and will be studied in future works.

[9]For the *ImageNet* experiment, the training loss is measured on a subset of the training data of size 50,000.

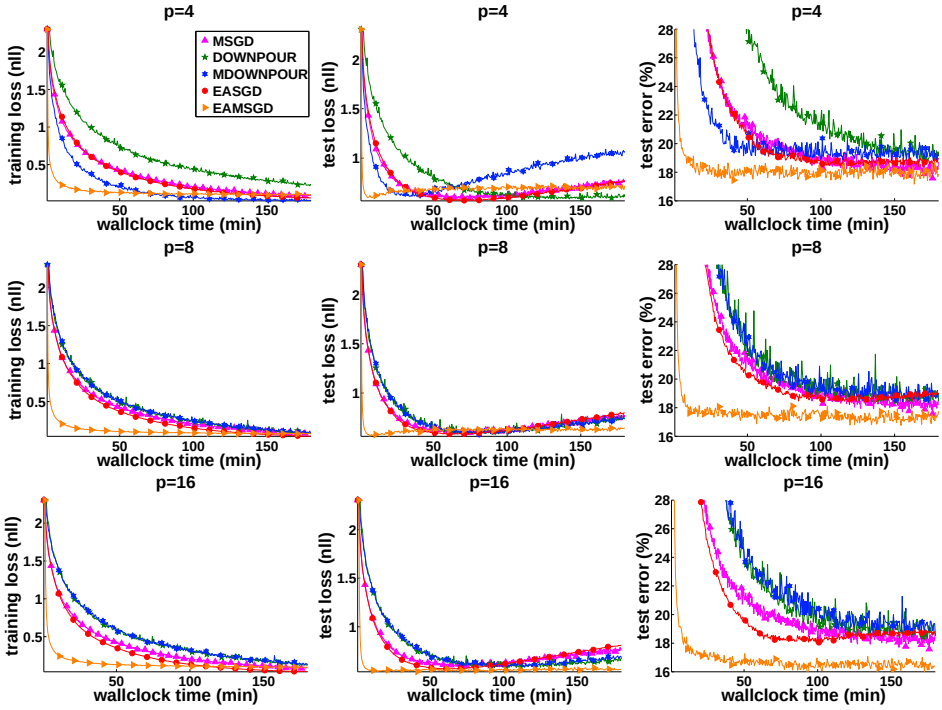

Figure 3: Training and test loss and the test error for the center variable versus a wallclock time for different number of local workers p for parallel methods (*MSGD* uses $p = 1$) on *CIFAR* with the 7-layer convolutional neural network. *EAMSGD* achieves significant accelerations compared to other methods, e.g. the relative speed-up for $p = 16$ (the best comparator method is then *MSGD*) to achieve the test error 21% equals 11.1.

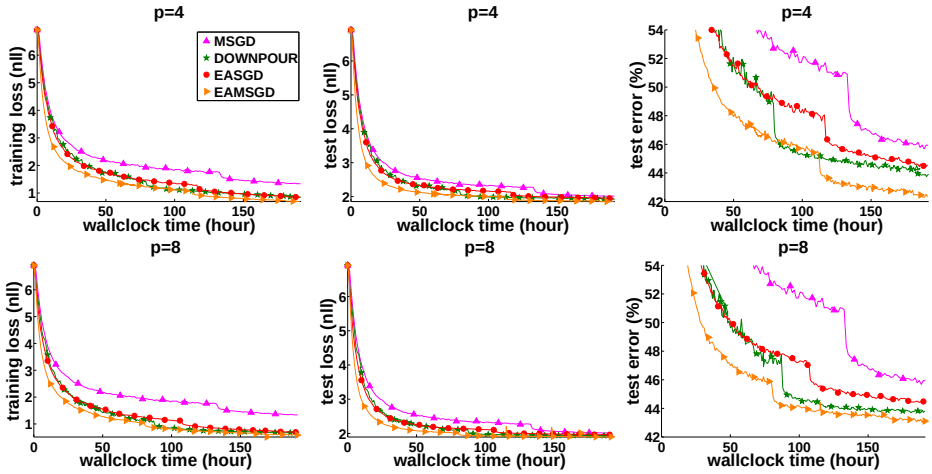

Figure 4: Training and test loss and the test error for the center variable versus a wallclock time for different number of local workers p (*MSGD* uses $p = 1$) on *ImageNet* with the 11-layer convolutional neural network. Initial learning rate is decreased twice, by a factor of 5 and then 2, when we observe that the online predictive loss [30] stagnates. *EAMSGD* achieves significant accelerations compared to other methods, e.g. the relative speed-up for $p = 8$ (the best comparator method is then *DOWNPOUR*) to achieve the test error 49% equals 1.8, and simultaneously it reduces the communication overhead (*DOWNPOUR* uses communication period $\tau = 1$ and *EAMSGD* uses $\tau = 10$).

Acknowledgments

The authors thank R. Power, J. Li for implementation guidance, J. Bruna, O. Henaff, C. Farabet, A. Szlam, Y. Bakhtin for helpful discussion, P. L. Combettes, S. Bengio and the referees for valuable feedback.

References

[1] Bottou, L. Online algorithms and stochastic approximations. In *Online Learning and Neural Networks*. Cambridge University Press, 1998.

[2] Dean, J, Corrado, G, Monga, R, Chen, K, Devin, M, Le, Q, Mao, M, Ranzato, M, Senior, A, Tucker, P, Yang, K, and Ng, A. Large scale distributed deep networks. In *NIPS*. 2012.

[3] Krizhevsky, A, Sutskever, I, and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.

[4] Sermanet, P, Eigen, D, Zhang, X, Mathieu, M, Fergus, R, and LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *ArXiv*, 2013.

[5] Nocedal, J and Wright, S. *Numerical Optimization, Second Edition*. Springer New York, 2006.

[6] Polyak, B. T and Juditsky, A. B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

[7] Bertsekas, D. P and Tsitsiklis, J. N. *Parallel and Distributed Computation*. Prentice Hall, 1989.

[8] Hestenes, M. R. *Optimization theory: the finite dimensional case*. Wiley, 1975.

[9] Boyd, S, Parikh, N, Chu, E, Peleato, B, and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.

[10] Shamir, O. Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *NIPS*. 2014.

[11] Yadan, O, Adams, K, Taigman, Y, and Ranzato, M. Multi-gpu training of convnets. In *Arxiv*. 2013.

[12] Paine, T, Jin, H, Yang, J, Lin, Z, and Huang, T. Gpu asynchronous stochastic gradient descent to speed up neural network training. In *Arxiv*. 2013.

[13] Seide, F, Fu, H, Droppo, J, Li, G, and Yu, D. 1-bit stochastic gradient descent and application to data-parallel distributed training of speech dnns. In *Interspeech 2014*, September 2014.

[14] Bekkerman, R, Bilenko, M, and Langford, J. Scaling up machine learning: Parallel and distributed approaches. Camridge Universityy Press, 2011.

[15] Choromanska, A, Henaff, M. B, Mathieu, M, Arous, G. B, and LeCun, Y. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

[16] Ho, Q, Cipar, J, Cui, H, Lee, S, Kim, J. K, Gibbons, P. B, Gibson, G. A, Ganger, G, and Xing, E. P. More effective distributed ml via a stale synchronous parallel parameter server. In *NIPS*. 2013.

[17] Azadi, S and Sra, S. Towards an optimal stochastic alternating direction method of multipliers. In *ICML*, 2014.

[18] Borkar, V. Asynchronous stochastic approximations. *SIAM Journal on Control and Optimization*, 36(3):840–851, 1998.

[19] Nedić, A, Bertsekas, D, and Borkar, V. Distributed asynchronous incremental subgradient methods. In *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications*, volume 8 of *Studies in Computational Mathematics*, pages 381 – 407. 2001.

[20] Langford, J, Smola, A, and Zinkevich, M. Slow learners are fast. In *NIPS*, 2009.

[21] Agarwal, A and Duchi, J. Distributed delayed stochastic optimization. In *NIPS*. 2011.

[22] Recht, B, Re, C, Wright, S. J, and Niu, F. Hogwild: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. In *NIPS*, 2011.

[23] Zinkevich, M, Weimer, M, Smola, A, and Li, L. Parallelized stochastic gradient descent. In *NIPS*, 2010.

[24] Nesterov, Y. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

[25] Lan, G. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

[26] Sutskever, I, Martens, J, Dahl, G, and Hinton, G. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

[27] Zhang, R and Kwok, J. Asynchronous distributed admm for consensus optimization. In *ICML*, 2014.

[28] Ouyang, H, He, N, Tran, L, and Gray, A. Stochastic alternating direction method of multipliers. In *Proceedings of the 30th International Conference on Machine Learning*, pages 80–88, 2013.

[29] Wan, L, Zeiler, M. D, Zhang, S, LeCun, Y, and Fergus, R. Regularization of neural networks using dropconnect. In *ICML*, 2013.

[30] Cesa-Bianchi, N, Conconi, A, and Gentile, C. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

[31] Nesterov, Y. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

Deep learning with Elastic Averaging SGD (Supplementary Material)

7 Additional theoretical results and proofs

7.1 Quadratic case

We provide here the convergence analysis of the synchronous *EASGD* algorithm with constant learning rate. The analysis is focused on the convergence of the center variable to the local optimum. We discuss one-dimensional quadratic case first, then the generalization to multi-dimensional setting (Lemma 7.3) and finally to the strongly convex case (Theorem 7.1).

Our analysis in the quadratic case extends the analysis of ASGD in [6]. Assume each of the p local workers $x_t^i \in \mathbb{R}^n$ observes a noisy gradient at time $t \geq 0$ of the linear form given in Equation 14.

$$g_t^i(x_t^i) = Ax_t^i - b - \xi_t^i, \quad i \in \{1, \ldots, p\}, \tag{14}$$

where the matrix A is positive-definite (each eigenvalue is strictly positive) and $\{\xi_t^i\}$'s are i.i.d. random variables, with zero mean and positive-definite covariance Σ. Let x^* denote the optimum solution, where $x^* = A^{-1}b \in \mathbb{R}^n$. In this section we analyze the behavior of the mean squared error (MSE) of the center variable $\tilde{x}_t$, where this error is denoted as $\mathbb{E}[\|\tilde{x}_t - x^*\|^2]$, as a function of t, p, η, α and β, where $\beta = p\alpha$. Note that the MSE error can be decomposed as (squared) bias and variance[10]: $\mathbb{E}[\|\tilde{x}_t - x^*\|^2] = \|\mathbb{E}[\tilde{x}_t - x^*]\|^2 + \mathbb{V}[\tilde{x}_t - x^*]$. For one-dimensional case ($n = 1$), we assume $A = h > 0$ and $\Sigma = \sigma^2 > 0$.

Lemma 7.1. *Let $\tilde{x}_0$ and $\{x_0^i\}_{i=1,\ldots,p}$ be arbitrary constants, then*

$$\mathbb{E}[\tilde{x}_t - x^*] = \gamma^t(\tilde{x}_0 - x^*) + \frac{\gamma^t - \phi^t}{\gamma - \phi}\alpha u_0, \tag{15}$$

$$\mathbb{V}[\tilde{x}_t - x^*] = \frac{p^2\alpha^2\eta^2}{(\gamma-\phi)^2}\left(\frac{\gamma^2 - \gamma^{2t}}{1-\gamma^2} + \frac{\phi^2 - \phi^{2t}}{1-\phi^2} - 2\frac{\gamma\phi - (\gamma\phi)^t}{1-\gamma\phi}\right)\frac{\sigma^2}{p}, \tag{16}$$

where $u_0 = \sum_{i=1}^p (x_0^i - x^* - \frac{\alpha}{1-p\alpha-\phi}(\tilde{x}_0 - x^*))$, $a = \eta h + (p+1)\alpha$, $c^2 = \eta h p\alpha$, $\gamma = 1 - \frac{a-\sqrt{a^2-4c^2}}{2}$, and $\phi = 1 - \frac{a+\sqrt{a^2-4c^2}}{2}$.

It follows from Lemma 7.1 that for the center variable to be stable the following has to hold

$$-1 < \phi < \gamma < 1. \tag{17}$$

It can be verified that ϕ and γ are the two zero-roots of the polynomial in λ: $\lambda^2 - (2-a)\lambda + (1 - a + c^2)$. Recall that ϕ and λ are the functions of η and α. Thus (see proof in Section 7.1.2)

- $\gamma < 1$ iff $c^2 > 0$ (i.e. $\eta > 0$ and $\alpha > 0$).
- $\phi > -1$ iff $(2 - \eta h)(2 - p\alpha) > 2\alpha$ and $(2 - \eta h) + (2 - p\alpha) > \alpha$.
- $\phi = \gamma$ iff $a^2 = 4c^2$ (i.e. $\eta h = \alpha = 0$).

The proof the above Lemma is based on the diagonalization of the linear gradient map (this map is symmetric due to the relation $\beta = p\alpha$). The stability analysis of the asynchronous *EASGD* algorithm in the round-robin scheme is similar due to this elastic symmetry.

Proof. Substituting the gradient from Equation 14 into the update rule used by each local worker in the synchronous *EASGD* algorithm (Equation 5 and 6) we obtain

$$x_{t+1}^i = x_t^i - \eta(Ax_t^i - b - \xi_t^i) - \alpha(x_t^i - \tilde{x}_t), \tag{18}$$

$$\tilde{x}_{t+1} = \tilde{x}_t + \sum_{i=1}^p \alpha(x_t^i - \tilde{x}_t), \tag{19}$$

[10] In our notation, $\mathbb{V}$ denotes the variance.

where η is the learning rate, and α is the moving rate. Recall that $\alpha = \eta\rho$ and $A = h$.

For the ease of notation we redefine $\tilde{x}_t$ and x_t^i as follows:

$$\tilde{x}_t \triangleq \tilde{x}_t - x^* \quad \text{and} \quad x_t^i \triangleq x_t^i - x^*.$$

We prove the lemma by explicitly solving the linear equations 18 and 19. Let $x_t = (x_t^1, \ldots, x_t^p, \tilde{x}_t)^T$. We rewrite the recursive relation captured in Equation 18 and 19 as simply

$$x_{t+1} = Mx_t + b_t,$$

where the drift matrix M is defined as

$$M = \begin{bmatrix} 1-\alpha-\eta h & 0 & ... & 0 & \alpha \\ 0 & 1-\alpha-\eta h & 0 & ... & \alpha \\ ... & 0 & ... & 0 & ... \\ 0 & ... & 0 & 1-\alpha-\eta h & \alpha \\ \alpha & \alpha & ... & \alpha & 1-p\alpha \end{bmatrix},$$

and the (diffusion) vector $b_t = (\eta\xi_t^1, \ldots, \eta\xi_t^p, 0)^T$.

Note that one of the eigenvalues of matrix M, that we call ϕ, satisfies $(1-\alpha-\eta h-\phi)(1-p\alpha-\phi) = p\alpha^2$. The corresponding eigenvector is $(1, 1, \ldots, 1, -\frac{p\alpha}{1-p\alpha-\phi})^T$. Let u_t be the projection of x_t onto this eigenvector. Thus $u_t = \sum_{i=1}^p (x_t^i - \frac{\alpha}{1-p\alpha-\phi}\tilde{x}_t)$. Let furthermore $\xi_t = \sum_{i=1}^p \xi_t^i$. Therefore we have

$$u_{t+1} = \phi u_t + \eta\xi_t. \tag{20}$$

By combining Equation 19 and 20 as follows

$$\begin{aligned} \tilde{x}_{t+1} &= \tilde{x}_t + \sum_{i=1}^p \alpha(x_t^i - \tilde{x}_t) = (1-p\alpha)\tilde{x}_t + \alpha(u_t + \frac{p\alpha}{1-p\alpha-\phi}\tilde{x}_t) \\ &= (1 - p\alpha + \frac{p\alpha^2}{1-p\alpha-\phi})\tilde{x}_t + \alpha u_t = \gamma\tilde{x}_t + \alpha u_t, \end{aligned}$$

where the last step results from the following relations: $\frac{p\alpha^2}{1-p\alpha-\phi} = 1-\alpha-\eta h-\phi$ and $\phi + \gamma = 1-\alpha-\eta h+1-p\alpha$. Thus we obtained

$$\tilde{x}_{t+1} = \gamma\tilde{x}_t + \alpha u_t. \tag{21}$$

Based on Equation 20 and 21, we can then expand u_t and $\tilde{x}_t$ recursively,

$$u_{t+1} = \phi^{t+1}u_0 + \phi^t(\eta\xi_0) + \ldots + \phi^0(\eta\xi_t), \tag{22}$$

$$\tilde{x}_{t+1} = \gamma^{t+1}\tilde{x}_0 + \gamma^t(\alpha u_0) + \ldots + \gamma^0(\alpha u_t). \tag{23}$$

Substituting $u_0, u_1, \ldots, u_t$, each given through Equation 22, into Equation 23 we obtain

$$\tilde{x}_t = \gamma^t\tilde{x}_0 + \frac{\gamma^t - \phi^t}{\gamma - \phi}\alpha u_0 + \alpha\eta\sum_{l=1}^{t-1}\frac{\gamma^{t-l} - \phi^{t-l}}{\gamma - \phi}\xi_{l-1}. \tag{24}$$

To be more specific, the Equation 24 is obtained by integrating by parts,

$$\begin{aligned} \tilde{x}_{t+1} &= \gamma^{t+1}\tilde{x}_0 + \sum_{i=0}^t \gamma^{t-i}(\alpha u_i) \\ &= \gamma^{t+1}\tilde{x}_0 + \sum_{i=0}^t \gamma^{t-i}(\alpha(\phi^i u_0 + \sum_{l=0}^{i-1}\phi^{i-1-l}\eta\xi_l)) \\ &= \gamma^{t+1}\tilde{x}_0 + \sum_{i=0}^t \gamma^{t-i}\phi^i(\alpha u_0) + \sum_{l=0}^{t-1}\sum_{i=l+1}^t \gamma^{t-i}\phi^{i-1-l}(\alpha\eta\xi_l) \\ &= \gamma^{t+1}\tilde{x}_0 + \frac{\gamma^{t+1} - \phi^{t+1}}{\gamma - \phi}(\alpha u_0) + \sum_{l=0}^{t-1}\frac{\gamma^{t-l} - \phi^{t-l}}{\gamma - \phi}(\alpha\eta\xi_l). \end{aligned}$$

Since the random variables ξ_l are i.i.d, we may sum the variance term by term as follows

$$\sum_{l=0}^{t-1}\left(\frac{\gamma^{t-l}-\phi^{t-l}}{\gamma-\phi}\right)^2 = \sum_{l=0}^{t-1}\frac{\gamma^{2(t-l)}-2\gamma^{t-l}\phi^{t-l}+\phi^{2(t-l)}}{(\gamma-\phi)^2}$$
$$= \frac{1}{(\gamma-\phi)^2}\left(\frac{\gamma^2-\gamma^{2(t+1)}}{1-\gamma^2}-2\frac{\gamma\phi-(\gamma\phi)^{t+1}}{1-\gamma\phi}+\frac{\phi^2-\phi^{2(t+1)}}{1-\phi^2}\right). \quad (25)$$

Note that $\mathbb{E}[\xi_t]=\sum_{i=1}^p \mathbb{E}[\xi_t^i]=0$ and $\mathbb{V}[\xi_t]=\sum_{i=1}^p \mathbb{V}[\xi_t^i]=p\sigma^2$. These two facts, the equality in Equation 24 and Equation 25 can then be used to compute $\mathbb{E}[\tilde{x}_t]$ and $\mathbb{V}[\tilde{x}_t]$ as given in Equation 15 and 16 in Lemma 7.1. □

7.1.1 Visualizing Lemma 7.1

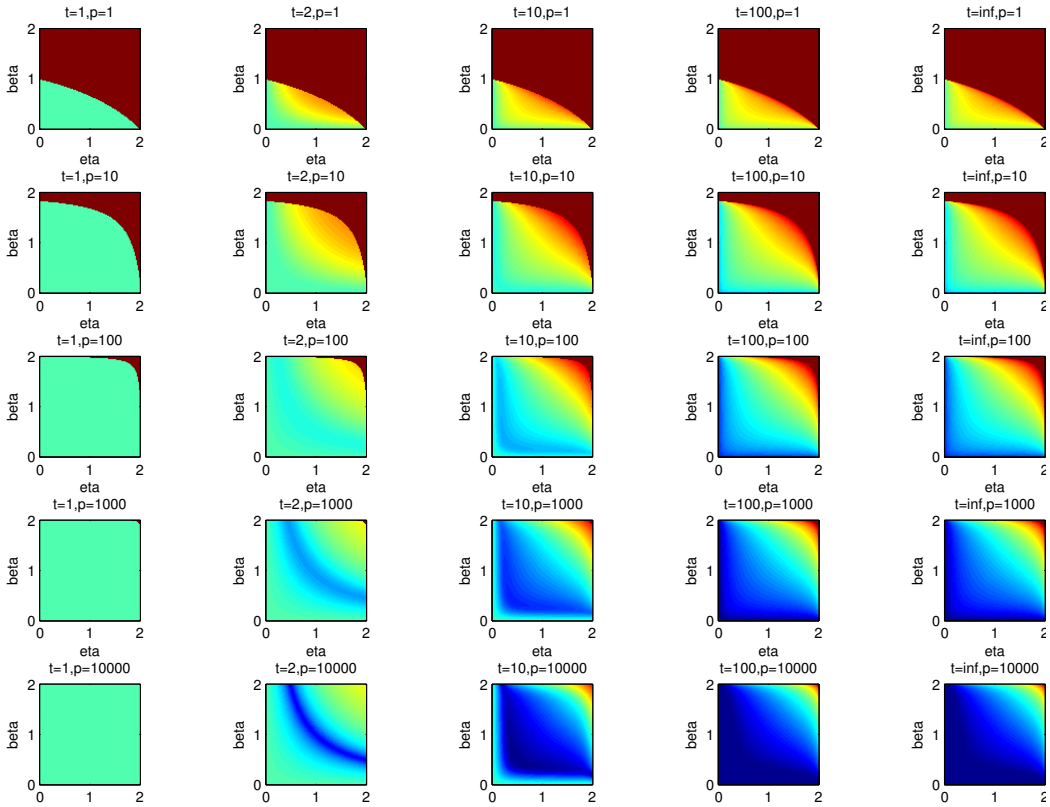

Figure 5: Theoretical mean squared error (MSE) of the center $\tilde{x}$ in the quadratic case, with various choices of the learning rate η (horizontal within each block), and the moving rate $\beta = p\alpha$ (vertical within each block), the number of processors $p = \{1, 10, 100, 1000, 10000\}$ (vertical across blocks), and the time steps $t = \{1, 2, 10, 100, \infty\}$ (horizontal across blocks). The MSE is plotted in log scale, ranging from 10^{-3} to 10^3 (from deep blue to red). The dark red (i.e. on the upper-right corners) indicates divergence.

In Figure 5, we illustrate the dependence of MSE on β, η and the number of processors p over time t. We consider the large-noise setting where $\tilde{x}_0 = x_0^i = 1$, $h = 1$ and $\sigma = 10$. The MSE error is color-coded such that the deep blue color corresponds to the MSE equal to 10^{-3}, the green color corresponds to the MSE equal to 1, the red color corresponds to MSE equal to 10^3 and the dark red color corresponds to the divergence of algorithm *EASGD* (condition in Equation 17 is then violated). The plot shows that we can achieve significant variance reduction by increasing the number of local workers p. This effect is less sensitive to the choice of β and η for large p.

7.1.2 Condition in Equation 17

We are going to show that

- $\gamma < 1$ iff $c^2 > 0$ (i.e. $\eta > 0$ and $\beta > 0$).
- $\phi > -1$ iff $(2-\eta h)(2-\beta) > 2\beta/p$ and $(2-\eta h) + (2-\beta) > \beta/p$.
- $\phi = \gamma$ iff $a^2 = 4c^2$ (i.e. $\eta h = \beta = 0$).

Recall that $a = \eta h + (p+1)\alpha$, $c^2 = \eta h p \alpha$, $\gamma = 1 - \frac{a-\sqrt{a^2-4c^2}}{2}$, $\phi = 1 - \frac{a+\sqrt{a^2-4c^2}}{2}$, and $\beta = p\alpha$. We have

- $\gamma < 1 \Leftrightarrow \frac{a-\sqrt{a^2-4c^2}}{2} > 0 \Leftrightarrow a > \sqrt{a^2-4c^2} \Leftrightarrow a^2 > a^2 - 4c^2 \Leftrightarrow c^2 > 0$.
- $\phi > -1 \Leftrightarrow 2 > \frac{a+\sqrt{a^2-4c^2}}{2} \Leftrightarrow 4 - a > \sqrt{a^2-4c^2} \Leftrightarrow 4 - a > 0, (4-a)^2 > a^2 - 4c^2 \Leftrightarrow 4 - a > 0, 4 - 2a + c^2 > 0 \Leftrightarrow 4 > \eta h + \beta + \alpha, 4 - 2(\eta h + \beta + \alpha) + \eta h \beta > 0$.
- $\phi = \gamma \Leftrightarrow \sqrt{a^2-4c^2} = 0 \Leftrightarrow a^2 = 4c^2$.

The next corollary is a consequence of Lemma 7.1. As the number of workers p grows, the averaging property of the *EASGD* can be characterized as follows

Corollary 7.1. *Let the Elastic Averaging relation $\beta = p\alpha$ and the condition 17 hold, then*

$$\lim_{p\to\infty}\lim_{t\to\infty} p\mathbb{E}[(\tilde{x}_t - x^*)^2] = \frac{\beta\eta h}{(2-\beta)(2-\eta h)} \cdot \frac{2-\beta-\eta h+\beta\eta h}{\beta+\eta h-\beta\eta h} \cdot \frac{\sigma^2}{h^2}.$$

Proof. Note that when β is fixed, $\lim_{p\to\infty} a = \eta h + \beta$ and $c^2 = \eta h \beta$. Then $\lim_{p\to\infty}\phi = \min(1-\beta, 1-\eta h)$ and $\lim_{p\to\infty}\gamma = \max(1-\beta, 1-\eta h)$. Also note that using Lemma 7.1 we obtain

$$\begin{aligned}
\lim_{t\to\infty}\mathbb{E}[(\tilde{x}_t - x^*)^2] &= \frac{\beta^2\eta^2}{(\gamma-\phi)^2}\left(\frac{\gamma^2}{1-\gamma^2} + \frac{\phi^2}{1-\phi^2} - \frac{2\gamma\phi}{1-\gamma\phi}\right)\frac{\sigma^2}{p} \\
&= \frac{\beta^2\eta^2}{(\gamma-\phi)^2}\left(\frac{\gamma^2(1-\phi^2)(1-\phi\gamma) + \phi^2(1-\gamma^2)(1-\phi\gamma) - 2\gamma\phi(1-\gamma^2)(1-\phi^2)}{(1-\gamma^2)(1-\phi^2)(1-\gamma\phi)}\right)\frac{\sigma^2}{p} \\
&= \frac{\beta^2\eta^2}{(\gamma-\phi)^2}\left(\frac{(\gamma-\phi)^2(1+\gamma\phi)}{(1-\gamma^2)(1-\phi^2)(1-\gamma\phi)}\right)\frac{\sigma^2}{p} \\
&= \frac{\beta^2\eta^2}{(1-\gamma^2)(1-\phi^2)} \cdot \frac{1+\gamma\phi}{1-\gamma\phi} \cdot \frac{\sigma^2}{p}.
\end{aligned}$$

Corollary 7.1 is obtained by plugining in the limiting values of ϕ and γ. □

The crucial point of Corollary 7.1 is that the MSE in the limit $t \to \infty$ is in the order of $1/p$ which implies that as the number of processors p grows, the MSE will decrease for the *EASGD* algorithm. Also note that the smaller the β is (recall that $\beta = p\alpha = p\eta\rho$), the more exploration is allowed (small ρ) and simultaneously the smaller the MSE is.

7.2 Generalization to multidimensional case

The next lemma (Lemma 7.2) shows that *EASGD* algorithm achieves the highest possible rate of convergence when we consider the double averaging sequence (similarly to [6]) $\{z_1, z_2, \dots\}$ defined as below

$$z_{t+1} = \frac{1}{t+1}\sum_{k=0}^{t}\tilde{x}_k. \tag{26}$$

Lemma 7.2 (Weak convergence)**.** *If the condition in Equation 17 holds, then the normalized double averaging sequence defined in Equation 26 converges weakly to the normal distribution with zero mean and variance σ^2/ph^2,*

$$\sqrt{t}(z_t - x^*) \rightharpoonup \mathcal{N}(0, \frac{\sigma^2}{ph^2}), \quad t \to \infty. \tag{27}$$

Proof. As in the proof of Lemma 7.1, for the ease of notation we redefine $\tilde{x}_t$ and x_t^i as follows:

$$\tilde{x}_t \triangleq \tilde{x}_t - x^* \quad \text{and} \quad x_t^i \triangleq x_t^i - x^*.$$

Also recall that $\{\xi_t^i\}$'s are *i.i.d.* random variables (noise) with zero mean and the same covariance $\Sigma \succ 0$. We are interested in the asymptotic behavior of the double averaging sequence $\{z_1, z_2, \dots\}$ defined as

$$z_{t+1} = \frac{1}{t+1}\sum_{k=0}^{t} \tilde{x}_k. \tag{28}$$

Recall the Equation 24 from the proof of Lemma 7.1 (for the convenience it is provided below):

$$\tilde{x}_k = \gamma^k \tilde{x}_0 + \alpha u_0 \frac{\gamma^k - \phi^k}{\gamma - \phi} + \alpha\eta \sum_{l=1}^{k-1} \frac{\gamma^{k-l} - \phi^{k-l}}{\gamma - \phi}\xi_{l-1},$$

where $\xi_t = \sum_{i=1}^p \xi_t^i$. Therefore

$$\begin{aligned}\sum_{k=0}^{t} \tilde{x}_k &= \frac{1-\gamma^{t+1}}{1-\gamma}\tilde{x}_0 + \alpha u_0 \frac{1}{\gamma-\mu}\left(\frac{1-\gamma^{t+1}}{1-\gamma} - \frac{1-\phi^{t+1}}{1-\phi}\right) + \alpha\eta \sum_{l=1}^{t-1}\sum_{k=l+1}^{t} \frac{\gamma^{k-l}-\phi^{k-l}}{\gamma-\phi}\xi_{l-1} \\ &= O(1) + \alpha\eta \sum_{l=1}^{t-1} \frac{1}{\gamma-\phi}\left(\gamma\frac{1-\gamma^{t-l}}{1-\gamma} - \phi\frac{1-\phi^{t-l}}{1-\phi}\right)\xi_{l-1}\end{aligned}$$

Note that the only non-vanishing term (in weak convergence) of $1/\sqrt{t}\sum_{k=0}^{t}\tilde{x}_k$ as $t \to \infty$ is

$$\frac{1}{\sqrt{t}}\alpha\eta \sum_{l=1}^{t-1} \frac{1}{\gamma-\phi}\left(\frac{\gamma}{1-\gamma} - \frac{\phi}{1-\phi}\right)\xi_{l-1}. \tag{29}$$

Also recall that $\mathbb{V}[\xi_{l-1}] = p\sigma^2$ and

$$\frac{1}{\gamma-\phi}\left(\frac{\gamma}{1-\gamma} - \frac{\phi}{1-\phi}\right) = \frac{1}{(1-\gamma)(1-\phi)} = \frac{1}{\eta h p \alpha}.$$

Therefore the expression in Equation 29 is asymptotically normal with zero mean and variance σ^2/ph^2. □

The asymptotic variance in the Lemma 7.2 is optimal with any fixed η and β for which Equation 17 holds. The next lemma (Lemma 7.3) extends the result in Lemma 7.2 to the multi-dimensional setting.

Lemma 7.3 (Weak convergence)**.** *Let h denotes the largest eigenvalue of A. If $(2-\eta h)(2-\beta) > 2\beta/p$, $(2-\eta h) + (2-\beta) > \beta/p$, $\eta > 0$ and $\beta > 0$, then the normalized double averaging sequence converges weakly to the normal distribution with zero mean and the covariance matrix $V = A^{-1}\Sigma(A^{-1})^T$,*

$$\sqrt{tp}(z_t - x^*) \rightharpoonup \mathcal{N}(0, V), \quad t \to \infty. \tag{30}$$

Proof. Since A is symmetric, one can use the proof technique of Lemma 7.2 to prove Lemma 7.3 by diagonalizing the matrix A. This diagonalization essentially generalizes Lemma 7.1 to the multidimensional case. We will not go into the details of this proof as we will provide a simpler way to look at the system. As in the proof of Lemma 7.1 and Lemma 7.2, for the ease of notation we redefine $\tilde{x}_t$ and x_t^i as follows:

$$\tilde{x}_t \triangleq \tilde{x}_t - x^* \quad \text{and} \quad x_t^i \triangleq x_t^i - x^*.$$

Let the spatial average of the local parameters at time t be denoted as y_t where $y_t = \frac{1}{p}\sum_{i=1}^p x_t^i$, and let the average noise be denoted as ξ_t, where $\xi_t = \frac{1}{p}\sum_{i=1}^p \xi_t^i$. Equations 18 and 19 can then be reduced to the following

$$y_{t+1} = y_t - \eta(Ay_t - \xi_t) + \alpha(\tilde{x}_t - y_t), \tag{31}$$

$$\tilde{x}_{t+1} = \tilde{x}_t + \beta(y_t - \tilde{x}_t). \tag{32}$$

We focus on the case where the learning rate η and the moving rate α are kept constant over time[11]. Recall $\beta = p\alpha$ and $\alpha = \eta\rho$.

Let's introduce the block notation $U_t = (y_t, \tilde{x}_t)$, $\Xi_t = (\eta\xi_t, 0)$, $M = I - \eta L$ and

$$L = \begin{pmatrix} A + \frac{\alpha}{\eta} I & -\frac{\alpha}{\eta} I \\ -\frac{\beta}{\eta} I & \frac{\beta}{\eta} I \end{pmatrix}.$$

From Equations 31 and 32 it follows that $U_{t+1} = MU_t + \Xi_t$. Note that this linear system has a degenerate noise Ξ_t which prevents us from directly applying results of [6]. Expanding this recursive relation and summing by parts, we have

$$\begin{aligned} \sum_{k=0}^{t} U_k &= M^0 U_0 + \\ & M^1 U_0 + M^0 \Xi_0 + \\ & M^2 U_0 + M^1 \Xi_0 + M^0 \Xi_1 + \\ & \dots \\ & M^t U_0 + M^{t-1} \Xi_0 + \cdots + M^0 \Xi_{t-1}. \end{aligned}$$

By Lemma 7.4, $\|M\|_2 < 1$ and thus

$$M^0 + M^1 + \cdots + M^t + \cdots = (I - M)^{-1} = \eta^{-1} L^{-1}.$$

Since A is invertible, we get

$$L^{-1} = \begin{pmatrix} A^{-1} & \frac{\alpha}{\beta} A^{-1} \\ A^{-1} & \frac{\eta}{\beta} + \frac{\alpha}{\beta} A^{-1} \end{pmatrix},$$

thus

$$\frac{1}{\sqrt{t}} \sum_{k=0}^{t} U_k = \frac{1}{\sqrt{t}} U_0 + \frac{1}{\sqrt{t}} \eta L^{-1} \sum_{k=1}^{t} \Xi_{k-1} - \frac{1}{\sqrt{t}} \sum_{k=1}^{t} M^{k+1} \Xi_{k-1}.$$

Note that the only non-vanishing term (in weak convergence) of $\frac{1}{\sqrt{t}} \sum_{k=0}^{t} U_k$ is $\frac{1}{\sqrt{t}} (\eta L)^{-1} \sum_{k=1}^{t} \Xi_{k-1}$ thus we have

$$\frac{1}{\sqrt{t}} (\eta L)^{-1} \sum_{k=1}^{t} \Xi_{k-1} \rightharpoonup \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} V & V \\ V & V \end{pmatrix} \right), \tag{33}$$

where $V = A^{-1} \Sigma (A^{-1})^T$. □

Lemma 7.4. *If the following conditions hold:*

$$\begin{aligned} (2 - \eta h)(2 - p\alpha) &> 2\alpha \\ (2 - \eta h) + (2 - p\alpha) &> \alpha \\ \eta &> 0 \\ \alpha &> 0 \end{aligned}$$

then $\|M\|_2 < 1$.

Proof. The eigenvalue λ of M and the (non-zero) eigenvector (y, z) of M satisfy

$$M \begin{pmatrix} y \\ z \end{pmatrix} = \lambda \begin{pmatrix} y \\ z \end{pmatrix}. \tag{34}$$

[11] As a side note, notice that the center parameter $\tilde{x}_t$ is tracking the spatial average y_t of the local parameters with a non-symmetric spring in Equation 31 and 32. To be more precise note that the update on y_{t+1} contains $(\tilde{x}_t - y_t)$ scaled by α, whereas the update on $\tilde{x}_{t+1}$ contains $-(\tilde{x}_t - y_t)$ scaled by β. Since $\alpha = \beta/p$ the impact of the center $\tilde{x}_{t+1}$ on the spatial local average y_{t+1} becomes more negligible as p grows.

Recall that

$$M = I - \eta L = \begin{pmatrix} I - \eta A - \alpha I & \alpha I \\ \beta I & I - \beta I \end{pmatrix}. \tag{35}$$

From the Equations 34 and 35 we obtain

$$\begin{cases} y - \eta A y - \alpha y + \alpha z = \lambda y \\ \beta y + (1-\beta) z = \lambda z \end{cases}. \tag{36}$$

Since (y, z) is assumed to be non-zero, we can write $z = \beta y/(\lambda + \beta - 1)$. Then the Equation 36 can be reduced to

$$\eta A y = (1 - \alpha - \lambda) y + \frac{\alpha\beta}{\lambda + \beta - 1} y. \tag{37}$$

Thus y is the eigenvector of A. Let λ_A be the eigenvalue of matrix A such that $Ay = \lambda_A y$. Thus based on Equation 37 it follows that

$$\eta \lambda_A = (1 - \alpha - \lambda) + \frac{\alpha\beta}{\lambda + \beta - 1}. \tag{38}$$

Equation 38 is equivalent to

$$\lambda^2 - (2 - a)\lambda + (1 - a + c^2) = 0, \tag{39}$$

where $a = \eta\lambda_A + (p+1)\alpha$, $c^2 = \eta\lambda_A p\alpha$. It follows from the condition in Equation 17 that $-1 < \lambda < 1$ iff $\eta > 0$, $\beta > 0$, $(2 - \eta\lambda_A)(2 - \beta) > 2\beta/p$ and $(2 - \eta\lambda_A) + (2 - \beta) > \beta/p$. Let h denote the maximum eigenvalue of A and note that $2 - \eta\lambda_A \geq 2 - \eta h$. This implies that the condition of our lemma is sufficient. □

As in Lemma 7.2, the asymptotic covariance in the Lemma 7.3 is optimal, i.e. meets the Fisher information lower-bound. The fact that this asymptotic covariance matrix V does not contain any term involving ρ is quite remarkable, since the penalty term ρ does have an impact on the condition number of the Hessian in Equation 2.

7.3 Strongly convex case

We now extend the above proof ideas to analyze the strongly convex case, in which the noisy gradient $g_t^i(x) = \nabla F(x) - \xi_t^i$ has the regularity that there exists some $0 < \mu \leq L$, for which $\mu \|x - y\|^2 \leq \langle \nabla F(x) - \nabla F(y), x - y \rangle \leq L \|x - y\|^2$ holds uniformly for any $x \in \mathbb{R}^d, y \in \mathbb{R}^d$. The noise $\{\xi_t^i\}$'s is assumed to be i.i.d. with zero mean and bounded variance $\mathbb{E}[\|\xi_t^i\|^2] \leq \sigma^2$.

Theorem 7.1. *Let* $a_t = \mathbb{E}\left\|\frac{1}{p}\sum_{i=1}^p x_t^i - x^*\right\|^2$, $b_t = \frac{1}{p}\sum_{i=1}^p \mathbb{E}\left\|x_t^i - x^*\right\|^2$, $c_t = \mathbb{E}\|\tilde{x}_t - x^*\|^2$, $\gamma_1 = 2\eta\frac{\mu L}{\mu + L}$ *and* $\gamma_2 = 2\eta L(1 - \frac{2\sqrt{\mu L}}{\mu + L})$. *If* $0 \leq \eta \leq \frac{2}{\mu + L}(1 - \alpha)$, $0 \leq \alpha < 1$ *and* $0 \leq \beta \leq 1$ *then*

$$\begin{pmatrix} a_{t+1} \\ b_{t+1} \\ c_{t+1} \end{pmatrix} \leq \begin{pmatrix} 1 - \gamma_1 - \gamma_2 - \alpha & \gamma_2 & \alpha \\ 0 & 1 - \gamma_1 - \alpha & \alpha \\ \beta & 0 & 1 - \beta \end{pmatrix} \begin{pmatrix} a_t \\ b_t \\ c_t \end{pmatrix} + \begin{pmatrix} \eta^2 \frac{\sigma^2}{p} \\ \eta^2 \sigma^2 \\ 0 \end{pmatrix}.$$

Proof. The idea of the proof is based on the point of view in Lemma 7.3, i.e. how close the center variable $\tilde{x}_t$ is to the spatial average of the local variables $y_t = \frac{1}{p}\sum_{i=1}^p x_t^i$. To further simplify the notation, let the noisy gradient be $\nabla f_{t,\xi}^i = g_t^i(x_t^i) = \nabla F(x_t^i) - \xi_t^i$, and $\nabla f_t^i = \nabla F(x_t^i)$ be its deterministic part. Then *EASGD* updates can be rewritten as follows,

$$x_{t+1}^i = x_t^i - \eta \nabla f_{t,\xi}^i - \alpha(x_t^i - \tilde{x}_t), \tag{40}$$

$$\tilde{x}_{t+1} = \tilde{x}_t + \beta(y_t - \tilde{x}_t). \tag{41}$$

We have thus the update for the spatial average,

$$y_{t+1} = y_t - \eta \frac{1}{p} \sum_{i=1}^p \nabla f_{t,\xi}^i - \alpha(y_t - \tilde{x}_t). \tag{42}$$

The idea of the proof is to bound the distance $\|\tilde{x}_t - x^*\|^2$ through $\|y_t - x^*\|^2$ and $\frac{1}{p}\sum_i^p \left\|x_t^i - x^*\right\|^2$. W start from the following estimate for the strongly convex function [31],

$$\langle \nabla F(x) - \nabla F(y), x - y\rangle \geq \frac{\mu L}{\mu + L}\|x - y\|^2 + \frac{1}{\mu + L}\|\nabla F(x) - \nabla F(y)\|^2 .$$

Since $\nabla f(x^*) = 0$, we have

$$\left\langle \nabla f_t^i, x_t^i - x^*\right\rangle \geq \frac{\mu L}{\mu + L}\left\|x_t^i - x^*\right\|^2 + \frac{1}{\mu + L}\left\|\nabla f_t^i\right\|^2 . \tag{43}$$

From Equation 40 the following relation holds,

$$\begin{aligned}\left\|x_{t+1}^i - x^*\right\|^2 &= \left\|x_t^i - x^*\right\|^2 + \eta^2\left\|\nabla f_{t,\xi}^i\right\|^2 + \alpha^2\left\|x_t^i - \tilde{x}_t\right\|^2 \\ &- 2\eta\left\langle \nabla f_{t,\xi}^i, x_t^i - x^*\right\rangle - 2\alpha\left\langle x_t^i - \tilde{x}_t, x_t^i - x^*\right\rangle \\ &+ 2\eta\alpha\left\langle \nabla f_{t,\xi}^i, x_t^i - \tilde{x}_t\right\rangle . \end{aligned}\tag{44}$$

By the cosine rule ($2\langle a - b, c - d\rangle = \|a - d\|^2 - \|a - c\|^2 + \|c - b\|^2 - \|d - b\|^2$), we have

$$2\left\langle x_t^i - \tilde{x}_t, x_t^i - x^*\right\rangle = \left\|x_t^i - x^*\right\|^2 + \left\|x_t^i - \tilde{x}_t\right\|^2 - \|\tilde{x}_t - x^*\|^2 . \tag{45}$$

By the Cauchy-Schwarz inequality, we have

$$\left\langle \nabla f_t^i, x_t^i - \tilde{x}_t\right\rangle \leq \left\|\nabla f_t^i\right\|\left\|x_t^i - \tilde{x}_t\right\| . \tag{46}$$

Combining the above estimates in Equations 43, 44, 45, 46, we obtain

$$\begin{aligned}\left\|x_{t+1}^i - x^*\right\|^2 &\leq \left\|x_t^i - x^*\right\|^2 + \eta^2\left\|\nabla f_t^i - \xi_t^i\right\|^2 + \alpha^2\left\|x_t^i - \tilde{x}_t\right\|^2 \\ &- 2\eta\left(\frac{\mu L}{\mu + L}\left\|x_t^i - x^*\right\|^2 + \frac{1}{\mu + L}\left\|\nabla f_t^i\right\|^2\right) + 2\eta\left\langle \xi_t^i, x_t^i - x^*\right\rangle \\ &- \alpha\left(\left\|x_t^i - x^*\right\|^2 + \left\|x_t^i - \tilde{x}_t\right\|^2 - \|\tilde{x}_t - x^*\|^2\right) \\ &+ 2\eta\alpha\left\|\nabla f_t^i\right\|\left\|x_t^i - \tilde{x}_t\right\| - 2\eta\alpha\left\langle \xi_t^i, x_t^i - \tilde{x}_t\right\rangle . \end{aligned}\tag{47}$$

Choosing $0 \leq \alpha < 1$, we can have this upper-bound for the terms $\alpha^2\left\|x_t^i - \tilde{x}_t\right\|^2 - \alpha\left\|x_t^i - \tilde{x}_t\right\|^2 + 2\eta\alpha\left\|\nabla f_t^i\right\|\left\|x_t^i - \tilde{x}_t\right\| = -\alpha(1 - \alpha)\left\|x_t^i - \tilde{x}_t\right\|^2 + 2\eta\alpha\left\|\nabla f_t^i\right\|\left\|x_t^i - \tilde{x}_t\right\| \leq \frac{\eta^2\alpha}{1-\alpha}\left\|\nabla f_t^i\right\|^2$ by applying $-ax^2 + bx \leq \frac{b^2}{4a}$ with $x = \left\|x_t^i - \tilde{x}_t\right\|$. Thus we can further bound Equation 47 with

$$\begin{aligned}\left\|x_{t+1}^i - x^*\right\|^2 &\leq (1 - 2\eta\frac{\mu L}{\mu + L} - \alpha)\left\|x_t^i - x^*\right\|^2 + (\eta^2 + \frac{\eta^2\alpha}{1 - \alpha} - \frac{2\eta}{\mu + L})\left\|\nabla f_t^i\right\|^2 \\ &- 2\eta^2\left\langle \nabla f_t^i, \xi_t^i\right\rangle + 2\eta\left\langle \xi_t^i, x_t^i - x^*\right\rangle - 2\eta\alpha\left\langle \xi_t^i, x_t^i - \tilde{x}_t\right\rangle \qquad (48) \\ &+ \eta^2\left\|\xi_t^i\right\|^2 + \alpha\|\tilde{x}_t - x^*\|^2 \qquad (49)\end{aligned}$$

As in Equation 48 and 49, the noise ξ_t^i is zero mean ($\mathbb{E}\xi_t^i = 0$) and the variance of the noise ξ_t^i is bounded ($\mathbb{E}\left\|\xi_t^i\right\|^2 \leq \sigma^2$), if η is chosen small enough such that $\eta^2 + \frac{\eta^2\alpha}{1-\alpha} - \frac{2\eta}{\mu+L} \leq 0$, then

$$\mathbb{E}\left\|x_{t+1}^i - x^*\right\|^2 \leq (1 - 2\eta\frac{\mu L}{\mu + L} - \alpha)\mathbb{E}\left\|x_t^i - x^*\right\|^2 + \eta^2\sigma^2 + \alpha\mathbb{E}\|\tilde{x}_t - x^*\|^2 . \tag{50}$$

Now we apply similar idea to estimate $\|y_t - x^*\|^2$. From Equation 42 the following relation holds,

$$\begin{aligned}\|y_{t+1} - x^*\|^2 &= \|y_t - x^*\|^2 + \eta^2\left\|\frac{1}{p}\sum_{i=1}^p \nabla f_{t,\xi}^i\right\|^2 + \alpha^2\|y_t - \tilde{x}_t\|^2 \\ &- 2\eta\left\langle \frac{1}{p}\sum_{i=1}^p \nabla f_{t,\xi}^i, y_t - x^*\right\rangle - 2\alpha\langle y_t - \tilde{x}_t, y_t - x^*\rangle \\ &+ 2\eta\alpha\left\langle \frac{1}{p}\sum_{i=1}^p \nabla f_{t,\xi}^i, y_t - \tilde{x}_t\right\rangle . \end{aligned}\tag{51}$$

By $\left\langle \frac{1}{p}\sum_{i=1}^{p} a_i, \frac{1}{p}\sum_{j=1}^{p} b_j \right\rangle = \frac{1}{p}\sum_{i=1}^{p}\langle a_i, b_i\rangle - \frac{1}{p^2}\sum_{i>j}\langle a_i - a_j, b_i - b_j\rangle$, we have

$$\left\langle \frac{1}{p}\sum_{i=1}^{p}\nabla f_t^i, y_t - x^* \right\rangle = \frac{1}{p}\sum_{i=1}^{p}\left\langle \nabla f_t^i, x_t^i - x^*\right\rangle - \frac{1}{p^2}\sum_{i>j}\left\langle \nabla f_t^i - \nabla f_t^j, x_t^i - x_t^j\right\rangle. \tag{52}$$

By the cosine rule, we have

$$2\left\langle y_t - \tilde{x}_t, y_t - x^*\right\rangle = \|y_t - x^*\|^2 + \|y_t - \tilde{x}_t\|^2 - \|\tilde{x}_t - x^*\|^2. \tag{53}$$

Denote $\xi_t = \frac{1}{p}\sum_{i=1}^{p}\xi_t^i$, we can rewrite Equation 51 as

$$\begin{aligned}
\|y_{t+1} - x^*\|^2 &= \|y_t - x^*\|^2 + \eta^2\left\|\frac{1}{p}\sum_{i=1}^{p}\nabla f_t^i - \xi_t\right\|^2 + \alpha^2\|y_t - \tilde{x}_t\|^2 \\
&- 2\eta\left\langle \frac{1}{p}\sum_{i=1}^{p}\nabla f_t^i - \xi_t, y_t - x^*\right\rangle - 2\alpha\left\langle y_t - \tilde{x}_t, y_t - x^*\right\rangle \\
&+ 2\eta\alpha\left\langle \frac{1}{p}\sum_{i=1}^{p}\nabla f_t^i - \xi_t, y_t - \tilde{x}_t\right\rangle. \qquad (54)
\end{aligned}$$

By combining the above Equations 52, 53 with 54, we obtain

$$\begin{aligned}
\|y_{t+1} - x^*\|^2 &= \|y_t - x^*\|^2 + \eta^2\left\|\frac{1}{p}\sum_{i=1}^{p}\nabla f_t^i - \xi_t\right\|^2 + \alpha^2\|y_t - \tilde{x}_t\|^2 \\
&- 2\eta\left(\frac{1}{p}\sum_{i=1}^{p}\left\langle \nabla f_t^i, x_t^i - x^*\right\rangle - \frac{1}{p^2}\sum_{i>j}\left\langle \nabla f_t^i - \nabla f_t^j, x_t^i - x_t^j\right\rangle\right) \qquad (55) \\
&+ 2\eta\left\langle \xi_t, y_t - x^*\right\rangle - \alpha(\|y_t - x^*\|^2 + \|y_t - \tilde{x}_t\|^2 - \|\tilde{x}_t - x^*\|^2) \\
&+ 2\eta\alpha\left\langle \frac{1}{p}\sum_{i=1}^{p}\nabla f_t^i - \xi_t, y_t - \tilde{x}_t\right\rangle. \qquad (56)
\end{aligned}$$

Thus it follows from Equation 43 and 56 that

$$\begin{aligned}
\|y_{t+1} - x^*\|^2 &\leq \|y_t - x^*\|^2 + \eta^2\left\|\frac{1}{p}\sum_{i=1}^{p}\nabla f_t^i - \xi_t\right\|^2 + \alpha^2\|y_t - \tilde{x}_t\|^2 \\
&- 2\eta\frac{1}{p}\sum_{i=1}^{p}\left(\frac{\mu L}{\mu + L}\left\|x_t^i - x^*\right\|^2 + \frac{1}{\mu + L}\left\|\nabla f_t^i\right\|^2\right) \\
&+ 2\eta\frac{1}{p^2}\sum_{i>j}\left\langle \nabla f_t^i - \nabla f_t^j, x_t^i - x_t^j\right\rangle \\
&+ 2\eta\left\langle \xi_t, y_t - x^*\right\rangle - \alpha(\|y_t - x^*\|^2 + \|y_t - \tilde{x}_t\|^2 - \|\tilde{x}_t - x^*\|^2) \\
&+ 2\eta\alpha\left\langle \frac{1}{p}\sum_{i=1}^{p}\nabla f_t^i - \xi_t, y_t - \tilde{x}_t\right\rangle. \qquad (57)
\end{aligned}$$

Recall $y_t = \frac{1}{p}\sum_{i=1}^{p} x_t^i$, we have the following bias-variance relation,

$$\begin{aligned}
\frac{1}{p}\sum_{i=1}^{p}\left\|x_t^i - x^*\right\|^2 &= \frac{1}{p}\sum_{i=1}^{p}\left\|x_t^i - y_t\right\|^2 + \|y_t - x^*\|^2 = \frac{1}{p^2}\sum_{i>j}\left\|x_t^i - x_t^j\right\|^2 + \|y_t - x^*\|^2, \\
\frac{1}{p}\sum_{i=1}^{p}\left\|\nabla f_t^i\right\|^2 &= \frac{1}{p^2}\sum_{i>j}\left\|\nabla f_t^i - \nabla f_t^j\right\|^2 + \left\|\frac{1}{p}\sum_{i=1}^{p}\nabla f_t^i\right\|^2. \qquad (58)
\end{aligned}$$

By the Cauchy-Schwarz inequality, we have

$$\frac{\mu L}{\mu+L}\left\|x_t^i - x_t^j\right\|^2 + \frac{1}{\mu+L}\left\|\nabla f_t^i - \nabla f_t^j\right\|^2 \geq \frac{2\sqrt{\mu L}}{\mu+L}\left\langle \nabla f_t^i - \nabla f_t^j, x_t^i - x_t^j\right\rangle. \tag{59}$$

Combining the above estimates in Equations 57, 58, 59, we obtain

$$\begin{aligned}
\|y_{t+1} - x^*\|^2 &\leq \|y_t - x^*\|^2 + \eta^2\left\|\frac{1}{p}\sum_{i=1}^p \nabla f_t^i - \xi_t\right\|^2 + \alpha^2\|y_t - \tilde{x}_t\|^2 \\
&- 2\eta\left(\frac{\mu L}{\mu+L}\|y_t - x^*\|^2 + \frac{1}{\mu+L}\left\|\frac{1}{p}\sum_{i=1}^p \nabla f_t^i\right\|^2\right) \\
&+ 2\eta\left(1 - \frac{2\sqrt{\mu L}}{\mu+L}\right)\frac{1}{p^2}\sum_{i>j}\left\langle \nabla f_t^i - \nabla f_t^j, x_t^i - x_t^j\right\rangle \\
&+ 2\eta\langle \xi_t, y_t - x^*\rangle - \alpha(\|y_t - x^*\|^2 + \|y_t - \tilde{x}_t\|^2 - \|\tilde{x}_t - x^*\|^2) \\
&+ 2\eta\alpha\left\langle \frac{1}{p}\sum_{i=1}^p \nabla f_t^i - \xi_t, y_t - \tilde{x}_t\right\rangle.
\end{aligned} \tag{60}$$

Similarly if $0 \leq \alpha < 1$, we can have this upper-bound for the terms $\alpha^2\|y_t - \tilde{x}_t\|^2 - \alpha\|y_t - \tilde{x}_t\|^2 + 2\eta\alpha\left\|\frac{1}{p}\sum_{i=1}^p \nabla f_t^i\right\|\|y_t - \tilde{x}_t\| \leq \frac{\eta^2\alpha}{1-\alpha}\left\|\frac{1}{p}\sum_{i=1}^p \nabla f_t^i\right\|^2$ by applying $-ax^2 + bx \leq \frac{b^2}{4a}$ with $x = \|y_t - \tilde{x}_t\|$. Thus we have the following bound for the Equation 60

$$\begin{aligned}
\|y_{t+1} - x^*\|^2 &\leq (1 - 2\eta\frac{\mu L}{\mu+L} - \alpha)\|y_t - x^*\|^2 + (\eta^2 + \frac{\eta^2\alpha}{1-\alpha} - \frac{2\eta}{\mu+L})\left\|\frac{1}{p}\sum_{i=1}^p \nabla f_t^i\right\|^2 \\
&- 2\eta^2\left\langle \frac{1}{p}\sum_{i=1}^p \nabla f_t^i, \xi_t\right\rangle + 2\eta\langle \xi_t, y_t - x^*\rangle - 2\eta\alpha\langle \xi_t, y_t - \tilde{x}_t\rangle \\
&+ 2\eta\left(1 - \frac{2\sqrt{\mu L}}{\mu+L}\right)\frac{1}{p^2}\sum_{i>j}\left\langle \nabla f_t^i - \nabla f_t^j, x_t^i - x_t^j\right\rangle \\
&+ \eta^2\|\xi_t\|^2 + \alpha\|\tilde{x}_t - x^*\|^2.
\end{aligned} \tag{61}$$

Since $\frac{2\sqrt{\mu L}}{\mu+L} \leq 1$, we need also bound the non-linear term $\left\langle \nabla f_t^i - \nabla f_t^j, x_t^i - x_t^j\right\rangle \leq L\left\|x_t^i - x_t^j\right\|^2$. Recall the bias-variance relation $\frac{1}{p}\sum_{i=1}^p\left\|x_t^i - x^*\right\|^2 = \frac{1}{p^2}\sum_{i>j}\left\|x_t^i - x_t^j\right\|^2 + \|y_t - x^*\|^2$. The key observation is that if $\frac{1}{p}\sum_{i=1}^p\left\|x_t^i - x^*\right\|^2$ remains bounded, then larger variance $\sum_{i>j}\left\|x_t^i - x_t^j\right\|^2$ implies smaller bias $\|y_t - x^*\|^2$. Thus this non-linear term can be compensated.

Again choose η small enough such that $\eta^2 + \frac{\eta^2\alpha}{1-\alpha} - \frac{2\eta}{\mu+L} \leq 0$ and take expectation in Equation 61,

$$\begin{aligned}
\mathbb{E}\|y_{t+1} - x^*\|^2 &\leq (1 - 2\eta\frac{\mu L}{\mu+L} - \alpha)\mathbb{E}\|y_t - x^*\|^2 \\
&+ 2\eta L\left(1 - \frac{2\sqrt{\mu L}}{\mu+L}\right)\left(\frac{1}{p}\sum_{i=1}^p \mathbb{E}\left\|x_t^i - x^*\right\|^2 - \mathbb{E}\|y_t - x^*\|^2\right) \\
&+ \eta^2\frac{\sigma^2}{p} + \alpha\mathbb{E}\|\tilde{x}_t - x^*\|^2.
\end{aligned} \tag{62}$$

As for the center variable in Equation 41, we apply simply the convexity of the norm $\|\cdot\|^2$ to obtain

$$\|\tilde{x}_{t+1} - x^*\|^2 \leq (1-\beta)\|\tilde{x}_t - x^*\|^2 + \beta\|y_t - x^*\|^2. \tag{63}$$

Combing the estimates from Equations 50, 62, 63, and denote $a_t = \mathbb{E}\left\|y_t - x^*\right\|^2$, $b_t = \frac{1}{p}\sum_{i=1}^{p}\mathbb{E}\left\|x_t^i - x^*\right\|^2$, $c_t = \mathbb{E}\left\|\tilde{x}_t - x^*\right\|^2$, $\gamma_1 = 2\eta\frac{\mu L}{\mu+L}$, $\gamma_2 = 2\eta L(1-\frac{2\sqrt{\mu L}}{\mu+L})$, then

$$\begin{pmatrix} a_{t+1} \\ b_{t+1} \\ c_{t+1} \end{pmatrix} \leq \begin{pmatrix} 1-\gamma_1-\gamma_2-\alpha & \gamma_2 & \alpha \\ 0 & 1-\gamma_1-\alpha & \alpha \\ \beta & 0 & 1-\beta \end{pmatrix} \begin{pmatrix} a_t \\ b_t \\ c_t \end{pmatrix} + \begin{pmatrix} \eta^2\frac{\sigma^2}{p} \\ \eta^2\sigma^2 \\ 0 \end{pmatrix},$$

as long as $0 \leq \beta \leq 1$, $0 \leq \alpha < 1$ and $\eta^2 + \frac{\eta^2\alpha}{1-\alpha} - \frac{2\eta}{\mu+L} \leq 0$, i.e. $0 \leq \eta \leq \frac{2}{\mu+L}(1-\alpha)$. □

8 Additional pseudo-codes of the algorithms

8.1 DOWNPOUR pseudo-code

Algorithm 3 captures the pseudo-code of the implementation of the DOWNPOUR used in this paper.

Algorithm 3: DOWNPOUR: Processing by worker i and the master

Input: learning rate η, communication period $\tau \in \mathbb{N}$
Initialize: $\tilde{x}$ is initialized randomly, $x^i = \tilde{x}$, $v^i = 0$, $t^i = 0$

Repeat
  if (τ divides t^i) **then**
    $\tilde{x} \leftarrow \tilde{x} + v^i$
    $x^i \leftarrow \tilde{x}$
    $v^i \leftarrow 0$
  end
  $x^i \leftarrow x^i - \eta g_{t^i}^i(x^i)$
  $v^i \leftarrow v^i - \eta g_{t^i}^i(x^i)$
  $t^i \leftarrow t^i + 1$
Until forever

8.2 MDOWNPOUR pseudo-code

Algorithms 4 and 5 capture the pseudo-codes of the implementation of momentum DOWNPOUR (MDOWNPOUR) used in this paper. Algorithm 4 shows the behavior of each local worker and Algorithm 5 shows the behavior of the master.

Algorithm 4: MDOWNPOUR: Processing by worker i

Initialize: $x^i = \tilde{x}$

Repeat
  Receive $\tilde{x}$ from the master: $x^i \leftarrow \tilde{x}$
  Compute gradient $g^i = g^i(x^i)$
  Send g^i to the master
Until forever

Algorithm 5: MDOWNPOUR: Processing by the master

Input: learning rate η, momentum term δ
Initialize: $\tilde{x}$ is initialized randomly, $v^i = 0$,

Repeat
  Receive g^i
  $v \leftarrow \delta v - \eta g^i$
  $\tilde{x} \leftarrow \tilde{x} + \delta v$
Until forever

9 Experiments - additional material

9.1 Data preprocessing

For the *ImageNet* experiment, we re-size each RGB image so that the smallest dimension is 256 pixels. We also re-scale each pixel value to the interval $[0, 1]$. We then extract random crops (and their horizontal flips) of size $3 \times 221 \times 221$ pixels and present these to the network in mini-batches of size 128.

For the *CIFAR* experiment, we use the original RGB image of size $3 \times 32 \times 32$. As before, we re-scale each pixel value to the interval $[0, 1]$. We then extract random crops (and their horizontal flips) of size $3 \times 28 \times 28$ pixels and present these to the network in mini-batches of size 128.

The training and test loss and the test error are only computed from the center patch ($3 \times 28 \times 28$) for the *CIFAR* experiment and the center patch ($3 \times 221 \times 221$) for the *ImageNet* experiment.

9.2 Data prefetching (Sampling the dataset by the local workers)

We will now explain precisely how the dataset is sampled by each local worker as uniformly and efficiently as possible. The general parallel data loading scheme on a single machine is as follows: we use k CPUs, where $k = 8$, to load the data in parallel. Each data loader reads from the memory-mapped (mmap) file a chunk of c raw images (preprocessing was described in the previous subsection) and their labels (for *CIFAR* $c = 512$ and for *ImageNet* $c = 64$). For the *CIFAR*, the mmap file of each data loader contains the entire dataset whereas for *ImageNet*, each mmap file of each data loader contains different $1/k$ fractions of the entire dataset. A chunk of data is always sent by one of the data loaders to the first worker who requests the data. The next worker requesting the data from the same data loader will get the next chunk. Each worker requests in total k data chunks from k different data loaders and then process them before asking for new data chunks. Notice that each data loader cycles through the data in the mmap file, sending consecutive chunks to the workers in order in which it receives requests from them. When the data loader reaches the end of the mmap file, it selects the address in memory uniformly at random from the interval $[0, s]$, where $s = (\text{number of images in the mmap file} \ \text{modulo} \ \text{mini-batch size})$, and uses this address to start cycling again through the data in the mmap file. After the local worker receives the k data chunks from the data loaders, it shuffles them and divides it into mini-batches of size 128.

9.3 Learning rates

In Table 1 we summarize the learning rates η (we used constant learning rates) explored for each method shown in Figure 2. For all values of τ the same set of learning rates was explored for each method.

Table 1: Learning rates explored for each method shown in Figure 2 (*CIFAR* experiment).

	η
EASGD	$\{0.05, 0.01, 0.005\}$
EAMSGD	$\{0.01, 0.005, 0.001\}$
DOWNPOUR ADOWNPOUR MVADOWNPOUR	$\{0.005, 0.001, 0.0005\}$
MDOWNPOUR	$\{0.00005, 0.00001, 0.000005\}$
SGD, ASGD, MVASGD	$\{0.05, 0.01, 0.005\}$
MSGD	$\{0.001, 0.0005, 0.0001\}$

In Table 2 we summarize the learning rates η (we used constant learning rates) explored for each method shown in Figure 3. For all values of p the same set of learning rates was explored for each method.

In Table 3 we summarize the initial learning rates η we use for each method shown in Figure 4. For all values of p the same set of learning rates was explored for each method. We also used the rule of the thumb to decrease the initial learning rate twice, first time we divided it by 5 and the second time by 2, when we observed that the decrease of the online predictive (training) loss saturates.

Table 2: Learning rates explored for each method shown in Figure 3 (*CIFAR* experiment).

	η
EASGD	$\{0.05, 0.01, 0.005\}$
EAMSGD	$\{0.01, 0.005, 0.001\}$
DOWNPOUR	$\{0.005, 0.001, 0.0005\}$
MDOWNPOUR	$\{0.00005, 0.00001, 0.000005\}$
SGD, ASGD, MVASGD	$\{0.05, 0.01, 0.005\}$
MSGD	$\{0.001, 0.0005, 0.0001\}$

Table 3: Learning rates explored for each method shown in Figure 4 (*ImageNet* experiment).

	η
EASGD	0.1
EAMSGD	0.001
DOWNPOUR	for $p = 4$: 0.02 for $p = 8$: 0.01
SGD, ASGD, MVASGD	0.05
MSGD	0.0005

9.4 Comparison of *SGD*, *ASGD*, *MVASGD* and *MSGD*

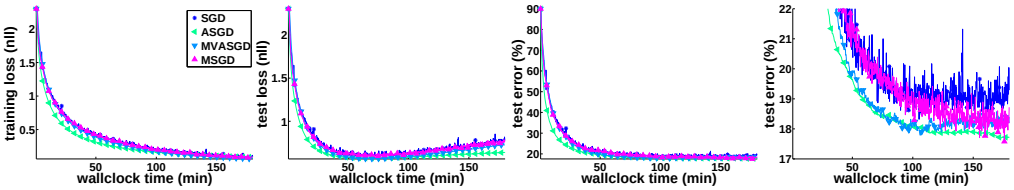

Figure 6: Convergence of the training and test loss (negative log-likelihood) and the test error (original and zoomed) computed for the center variable as a function of wallclock time for *SGD*, *ASGD*, *MVASGD* and *MSGD* ($p = 1$) on the *CIFAR* experiment.

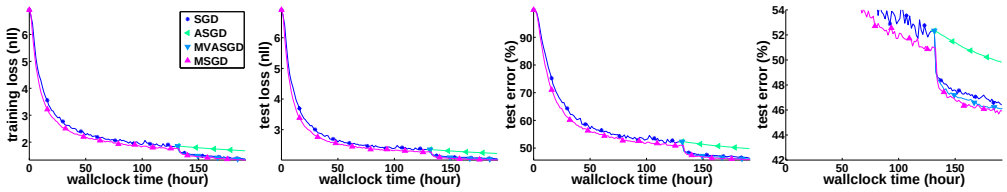

Figure 7: Convergence of the training and test loss (negative log-likelihood) and the test error (original and zoomed) computed for the center variable as a function of wallclock time for *SGD*, *ASGD*, *MVASGD* and *MSGD* ($p = 1$) on the *ImageNet* experiment.

Figure 6 shows the convergence of the training and test loss (negative log-likelihood) and the test error computed for the center variable as a function of wallclock time for *SGD*, *ASGD*, *MVASGD* and *MSGD* ($p = 1$) on the *CIFAR* experiment. For all *CIFAR* experiments we always start the averaging for the $ADOWNPOUR$ and $ASGD$ methods from the very beginning of each experiment. For all *ImageNet* experiments we start the averaging for the $ASGD$ at the same time when we first reduce the initial learning rate.

Figure 7 shows the convergence of the training and test loss (negative log-likelihood) and the test error computed for the center variable as a function of wallclock time for *SGD*, *ASGD*, *MVASGD* and *MSGD* ($p = 1$) on the *ImageNet* experiment.

9.5 Dependence of the learning rate

This section discusses the dependence of the trade-off between exploration and exploitation on the learning rate. We compare the performance of respectively *EAMSGD* and *EASGD* for different learning rates η when $p = 16$ and $\tau = 10$ on the *CIFAR* experiment. We observe in Figure 8 that higher learning rates η lead to better test performance for the *EAMSGD* algorithm which potentially can be justified by the fact that they sustain higher fluctuations of the local workers. We conjecture that higher fluctuations lead to more exploration and simultaneously they also impose higher regularization. This picture however seems to be opposite for the *EASGD* algorithm for which larger learning rates hurt the performance of the method and lead to overfitting. Interestingly in this experiment for both *EASGD* and *EAMSGD* algorithm, the learning rate for which the best training performance was achieved simultaneously led to the worst test performance.

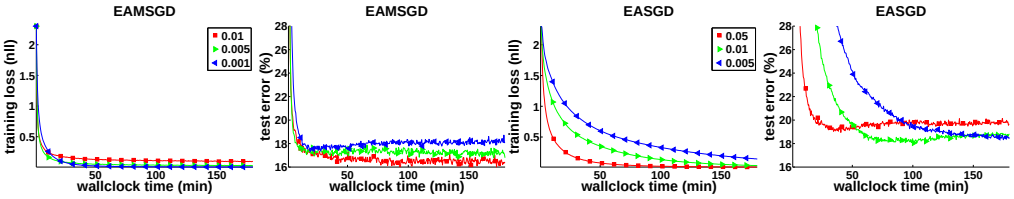

Figure 8: Convergence of the training loss (negative log-likelihood, original) and the test error (zoomed) computed for the center variable as a function of wallclock time for *EAMSGD* and *EASGD* run with different values of η on the *CIFAR* experiment. $p = 16$, $\tau = 10$.

9.6 Dependence of the communication period

This section discusses the dependence of the trade-off between exploration and exploitation on the communication period. We have observed from the *CIFAR* experiment that *EASGD* algorithm exhibits very similar convergence behavior when $\tau = 1$ up to even $\tau = 1000$, whereas *EAMSGD* can get trapped at worse energy (loss) level for $\tau = 100$. This behavior of *EAMSGD* is most likely due to the non-convexity of the objective function. Luckily, it can be avoided by gradually decreasing the learning rate, i.e. increasing the penalty term ρ (recall $\alpha = \eta\rho$), as shown in Figure 9. In contrast, the *EASGD* algorithm does not seem to get trapped at all along its trajectory. The performance of *EASGD* is less sensitive to increasing the communication period compared to *EAMSGD*, whereas for the *EAMSGD* the careful choice of the learning rate for large communication periods seems crucial.

Compared to all earlier results, the experiment in this section is re-run three times with a new random[12] seed and with faster cuDNN[13] package[14]. All our methods are implemented in Torch[15]. The Message Passing Interface implementation MVAPICH2[16] is used for the GPU-CPU communication.

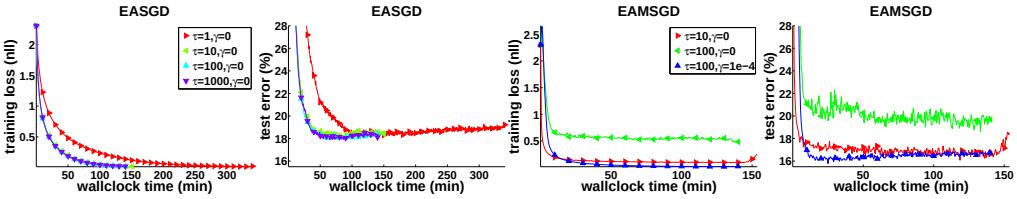

Figure 9: Convergence of the training loss (negative log-likelihood, original) and the test error (zoomed) computed for the center variable as a function of wallclock time for *EASGD* and *EAMSGD* ($p = 16, \eta = 0.01, \beta = 0.9, \delta = 0.99$) on the *CIFAR* experiment with various communication period τ and learning rate decay γ. The learning rate is decreased gradually over time based each local worker's own clock t with $\eta_t = \eta/(1 + \gamma t)^{0.5}$.

[12] To clarify, the random initialization we use is by default in Torch's implementation.

[13] `https://developer.nvidia.com/cuDNN`

[14] `https://github.com/soumith/cudnn.torch`

[15] `http://torch.ch`

[16] `http://mvapich.cse.ohio-state.edu`

9.7 Breakdown of the wallclock time

In addition, we report in Table 4 the breakdown of the total running time for *EASGD* when $\tau = 10$ (the time breakdown for *EAMSGD* is almost identical) and *DOWNPOUR* when $\tau = 1$ into computation time, data loading time and parameter communication time. For the *CIFAR* experiment the reported time corresponds to processing 400×128 data samples whereas for the *ImageNet* experiment it corresponds to processing 1024×128 data samples. For $\tau = 1$ and $p \in \{8, 16\}$ we observe that the communication time accounts for significant portion of the total running time whereas for $\tau = 10$ the communication time becomes negligible compared to the total running time (recall that based on previous results *EASGD* and *EAMSGD* achieve best performance with larger τ which is ideal in the setting when communication is time-consuming).

	$p = 1$	$p = 4$	$p = 8$	$p = 16$
$\tau = 1$	12/1/0	11/2/3	11/2/5	11/2/9
$\tau = 10$	NA	11/2/1	11/2/1	12/2/1

	$p = 1$	$p = 4$	$p = 8$
$\tau = 1$	1248/20/0	1323/24/173	1239/61/284
$\tau = 10$	NA	1254/58/7	1266/84/11

Table 4: Approximate computation time, data loading time and parameter communication time [sec] for *DOWNPOUR* (top line for $\tau = 1$) and *EASGD* (the time breakdown for *EAMSGD* is almost identical) (bottom line for $\tau = 10$). Left time corresponds to *CIFAR* experiment and right table corresponds to *ImageNet* experiment.

9.8 Time speed-up

In Figure 10 and 11, we summarize the wall clock time needed to achieve the same level of the test error for all the methods in the *CIFAR* and *ImageNet* experiment as a function of the number of local workers p. For the *CIFAR* (Figure 10) we examined the following levels: $\{21\%, 20\%, 19\%, 18\%\}$ and for the *ImageNet* (Figure 11) we examined: $\{49\%, 47\%, 45\%, 43\%\}$. If some method does not appear on the figure for a given test error level, it indicates that this method never achieved this level. For the *CIFAR* experiment we observe that from among *EASGD*, *DOWNPOUR* and *MDOWNPOUR* methods, the *EASGD* method needs less time to achieve a particular level of test error. We observe that with higher p each of these methods does not necessarily need less time to achieve the same level of test error. This seems counter intuitive though recall that the learning rate for the methods is selected based on the smallest achievable test error. For larger p smaller learning rates were selected than for smaller p which explains our results. Meanwhile, the *EAMSGD* method achieves significant speed-up over other methods for all the test error levels. For the *ImageNet* experiment we observe that all methods outperform *MSGD* and furthermore with $p = 4$ or $p = 8$ each of these methods requires less time to achieve the same level of test error. The *EAMSGD* consistently needs less time than any other method, in particular *DOWNPOUR*, to achieve any of the test error levels.

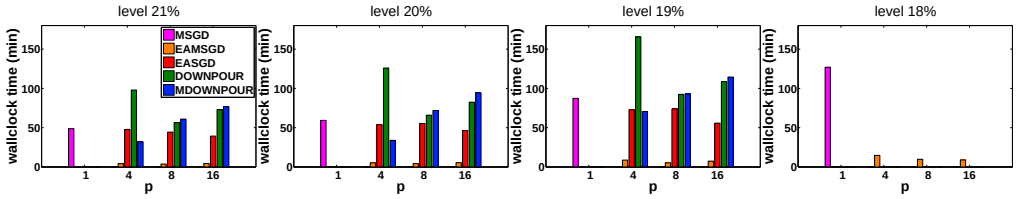

Figure 10: The wall clock time needed to achieve the same level of the test error thr as a function of the number of local workers p on the *CIFAR* dataset. From left to right: $thr = \{21\%, 20\%, 19\%, 18\%\}$. Missing bars denote that the method never achieved specified level of test error.

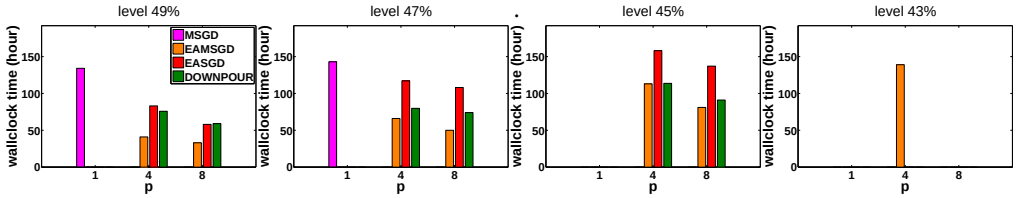

Figure 11: The wall clock time needed to achieve the same level of the test error thr as a function of the number of local workers p on the *ImageNet* dataset. From left to right: $thr = \{49\%, 47\%, 45\%, 43\%\}$. Missing bars denote that the method never achieved specified level of test error.

.