OPEN ACCESS

King's College London

King's Research Portal

DOI:
10.1109/LCOMM.2019.2922658

Document Version
Peer reviewed version

Link to publication record in King's Research Portal

Citation for published version (APA):
Skatchkovsky, N., & Simeone, O. (2019). Optimizing Pipelined Computation and Communication for Latency-Constrained Edge Learning. *IEEE COMMUNICATIONS LETTERS*, *23*(9), 1542-1546. Article 8736251. https://doi.org/10.1109/LCOMM.2019.2922658

Citing this paper
Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights
Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

•Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
•You may not further distribute the material or use it for any profit-making activity or commercial gain
•You may freely distribute the URL identifying the publication in the Research Portal

Take down policy
If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 27. Dec. 2024

arXiv:1906.04488v1 [cs.LG] 11 Jun 2019

OPTIMIZING PIPELINED COMPUTATION AND COMMUNICATION FOR LATENCY-CONSTRAINED EDGE LEARNING

A PREPRINT

Nicolas Skatchkovsky*
Department of Informatics
King's College London
United Kingdom
nicolas.skatchkovsky@kcl.ac.uk

Osvaldo Simeone*
Department of Informatics
King's College London
United Kingdom
osvaldo.simeone@kcl.ac.uk

June 12, 2019

ABSTRACT

Consider a device that is connected to an edge processor via a communication channel. The device holds local data that is to be offloaded to the edge processor so as to train a machine learning model, e.g., for regression or classification. Transmission of the data to the learning processor, as well as training based on Stochastic Gradient Descent (SGD), must be both completed within a time limit. Assuming that communication and computation can be pipelined, this work investigates the optimal choice for the packet payload size, given the overhead of each data packet transmission and the ratio between the computation and the communication rates. This amounts to a tradeoff between bias and variance, since communicating the entire data set first reduces the bias of the training process but it may not leave sufficient time for learning. Analytical bounds on the expected optimality gap are derived so as to enable an effective optimization, which is validated in numerical results.

Keywords Machine learning · Mobile Edge Computing · Stochastic Gradient Descent

1 Introduction

Edge learning refers to the training of machine learning models on devices that are close to the end users [1]. The proximity to the user is instrumental in facilitating a low-latency response, in enhancing privacy, and in reducing backhaul congestion. Edge learning processors include smart phones and other user-owned devices, as well as edge nodes of a wireless network that provide wireless access and computational resources [1]. As illustrated in Fig. 1, the latter case hinges on the offloading of data from the data-bearing device to the edge processor, and can be seen as an instance of mobile edge computing [2].

Research on edge learning has so far instead focused mostly on scenarios in which training occurs locally at the data-bearing devices. In these setups, devices can communicate either through a parameter server [3] or in a device-to-device manner [4]. The goal is to either learn a global model without exchanging directly the local data [5] or to train separate models while leveraging the correlation among the local data sets [6]. Devices can exchange either information about the local model parameters, as in federated learning [7], or gradient information, as in distributed Stochastic Gradient Descent (SGD) methods [8, 9].

*The authors have received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (Grant Agreement No. 725731).

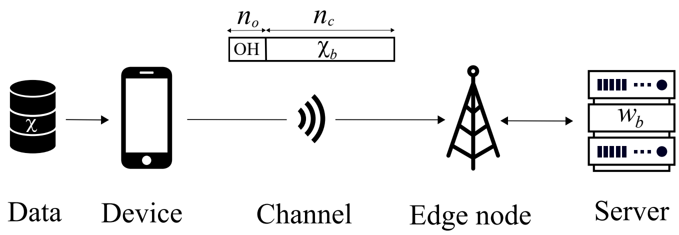

Figure 1: An edge computing system, in which training of a model parametrized by vector w takes place at an edge processor based on data received from a device using a protocol with timeline illustrated in Fig. 2 (OH = overhead).

In this work, we consider an edge learning scenario in which training takes place at an edge node of a wireless system as illustrated in Fig. 1. The data is held by a device and has to be offloaded through a communication channel to the edge node. The learning task has to be executed within a time limit, which might be insufficient to transmit the complete dataset. Transmission of data blocks from device to edge node, and training at the edge node can be carried out simultaneously (see Fig. 2). Each transmitted packet contains a fixed overhead, accounting e.g. for meta-data and pilots. Given the overhead of each data packet transmission, what is the optimal size of a communication block? Communicating the entire data set first reduces the bias of the training process but it may not leave sufficient time for learning. We investigate a more general strategy that communicates in blocks and pipelines communication and computation with an optimized block size, which is shown to be generally preferable. Analysis and simulation results provide insights into the optimal duration of the communication block and on the performance gains attainable with an optimized communication and computation policy.

The rest of this work is organized as follows. In Sec. 2, we provide an overview of the model and the associated notations. In Sec. 3, we examine the technical assumptions necessary for our work. In Sec. 4, we provide our main result and discuss its implications. Finally, in Sec. 5, we consider numerical experiments in the light of our result.

2 System model

As seen in Fig. 1, we study an edge learning system in which a device communicates with an edge node, and associated server, over an error-free communication channel. The device has access to a local training dataset $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ of N data points $\{x_n\}_{n=1}^N$, and training of a machine learning model is carried out at the edge node based on data received from the device. As illustrated in Fig. 2, communication and learning must be completed within a time limit T. To this end, the transmissions are organized into blocks, and transmission and computing at the edge node can be performed in parallel.

Training at the edge node aims at identifying a model parametrized by a vector $w \in \mathrm{R}^d$ within a given hypothesis class. Training is carried out by (approximately) solving the Empirical Risk Minimization (ERM) problem (see, e.g, [10]). This amounts to the minimization with respect to vector w of the empirical average $\mathcal{L}(w)$ of a loss function $\ell(w, x)$ over all the data points x in the training dataset, i.e.,

$$\mathcal{L}(w) = \frac{1}{N} \sum_{n=1}^{N} \ell(w, x_n). \tag{1}$$

As detailed below, the minimization of the function $\mathcal{L}(w)$ is carried out at the edge node using SGD, based on the data points received from the device.

In order to elaborate on the communication and computation protocol illustrated in Fig. 2, we normalize all time measures to the time required to transmit one data sample from the device to the edge node. With this convention, we denote as τ_p the time required to make one SGD update at the edge node.

As seen in Fig. 2, transmission from the device to the edge node is organised into blocks. In this study, we ignore the effect of channel errors, which is briefly discussed in Sec. 6. In the b-th block, the device transmits a subset $\mathcal{X}_b \subseteq \mathcal{X}$ of n_c new samples from its local dataset. At the end of the block, the edge node adds these samples to the subset $\tilde{\mathcal{X}}_{b+1}$ of samples it has available for training in the $b+1$-th block, i.e., $\tilde{\mathcal{X}}_{b+1} = \tilde{\mathcal{X}}_b \cup \mathcal{X}_b$ with $\mathcal{X}_0 = \varnothing$. The samples in $\mathcal{X}_b$ are randomly and uniformly selected from the set $\Delta\mathcal{X}_b = \mathcal{X} \setminus \tilde{\mathcal{X}}_b$ of samples not yet transmitted to the edge node. A packet

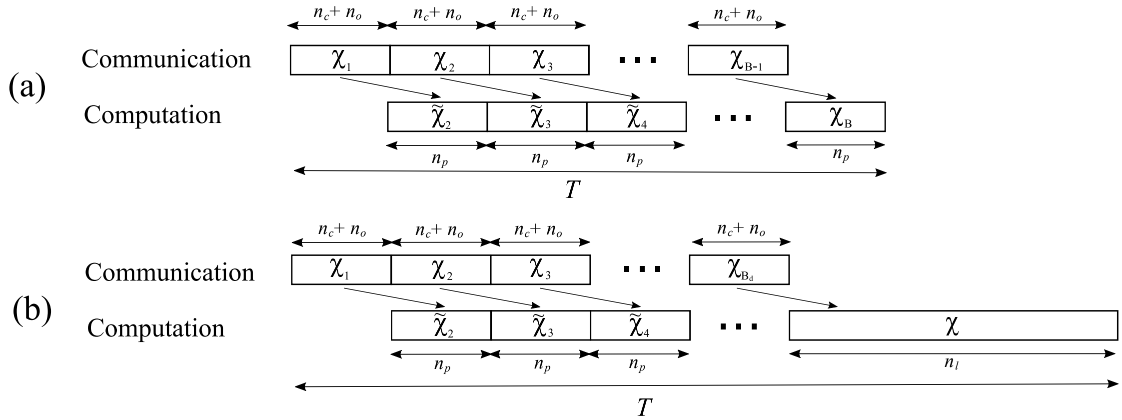

Figure 2: Transmission and training protocol: when (a) $T \le B_d(n_c + n_o)$; and (b) $T > B_d(n_c + n_o)$.

sent in any block contains an overhead, e.g., for pilots and meta-data, of duration n_o, irrespective of the number n_c of transmitted samples. It follows that the duration of a transmission block is $n_c + n_o$.

There are at most $B_d = N/n_c$ transmission blocks, since B_d blocks are sufficient to deliver the entire dataset to the edge node. Therefore, we need to distinguish two cases. As seen in Fig. 2(a), when $T \le B_d(n_c + n_o)$, the device is only able to deliver a fraction of the samples. In particular, denoting as $B = T/(n_c + n_o)$ the number of blocks, the fraction of data points delivered at the edge node at time T equals $(B-1)/B_d$. In contrast, if $T > B_d(n_c + n_o)$, as illustrated in Fig. 2(b), the edge node has the entire dataset available after B_d blocks, that is, for a duration equal to $\tau_l = T - B_d(n_c + n_o)$. Henceforth, we refer to this last period as block $B_l = B_d + 1$.

During each block $b \le B_d$, the edge node computes $n_p = (n_c + n_o)/\tau_p$ local SGD updates (2). During block B_l, the edge node computes $n_l = \tau_l/\tau_p$ SGD updates. The j-th local update at block b, with $j = 1, \dots, n_p$, is given as

$$w_b^j = w_b^{j-1} - \alpha \nabla \ell(w_b^{j-1}, \xi_b^j), \tag{2}$$

where α is the learning rate, and ξ_b^j is a data point sampled i.i.d. uniformly from the subset $\tilde{\mathcal{X}}_b = \bigcup_{l=1}^{b-1} \mathcal{X}_l$ of samples currently available at the edge node. Note that we have $\tilde{\mathcal{X}}_{B_l} = \mathcal{X}$.

The goal of this work is to optimize the number of samples n_c sent in each block with the aim of minimizing the empirical loss (1) at the edge node at the end of time T. In the next sections, we present an analysis of the empirical loss obtained at time T that allows us to gain insights into the optimal choice of n_c.

3 Technical assumptions

In order to study the training loss achieved at the edge node at the end of the training process, we make the following standard assumptions, which apply, for instance, to linear models with quadratic or cross-entropy losses under suitable constraints (see the comprehensive review paper [9]):

$(A1)$ the sequence of iterates w_b^j in (2) is contained in a bounded open set $\mathcal{W} \subseteq \mathbf{R}^d$ with radius $D = \max_{u,w \in \mathcal{W}\times\mathcal{W}} ||w - u||_2$ over which the function $\ell(w,x)$ is bounded below by a scalar ℓ_{inf} for all x;

$(A2)$ the function $\ell(w,x)$ is continuously differentiable in w for any fixed value of x and is L-smooth in w, i.e.,

$$||\nabla \ell(w,x) - \nabla \ell(\bar{w},x)||_2 \le L||w - \bar{w}||_2 \tag{3}$$

for all $(w,\bar{w}) \in \mathcal{W} \times \mathcal{W}$, and for all x. This implies

$$\ell(w,x) \le \ell(\bar{w},x) + \nabla \ell(\bar{w},x)^T (w - \bar{w}) + \frac{L}{2}||w - \bar{w}||_2^2 \tag{4}$$

for all $(w,\bar{w}) \in \mathcal{W} \times \mathcal{W}$, and for all x;

$(A3)$ the loss function $\ell(w,x)$ is convex and satisties the Polyak-Lojasiewicz condition in w, i.e., there exists a constant $c > 0$ such that

$$2c(\ell(w,x) - \ell(w_\ell^*,x)) \leq ||\nabla \ell(w,x)||_2^2 \tag{5}$$

for all $(w,x) \in \mathcal{W} \times \mathbf{R}^d$ where $w_\ell^*(x) = \arg\min_{w\in\mathcal{W}} \ell(w,x)$ is a minimizer of $\ell(w,x)$. The P-L condition is implied by, but does not imply, strong convexity [9].

We further need to make assumptions on the statistics of the gradient $\nabla \ell(w, \xi_b^j)$ used in the update (2). To this end, for each block $b > 1$, we define the empirical loss limited to the samples available at the edge node at block b as

$$\tilde{\mathcal{L}}_b(w) = \frac{1}{(b-1)n_c} \sum_{x_i \in \tilde{\mathcal{X}}_b} \ell(w,x_i); \tag{6}$$

the empirical loss over the samples transmitted at iteration $b \geq 1$ as

$$\mathcal{L}_b(w) = \frac{1}{n_c} \sum_{x_i \in \mathcal{X}_b} \ell(w,x_i); \tag{7}$$

and the empirical loss over the samples not available at the edge at iteration $b > 1$

$$\Delta\mathcal{L}_b(w) = \frac{1}{N-(b-1)n_c} \sum_{x_i \in \Delta\mathcal{X}_b} \ell(w,x_i). \tag{8}$$

Note that we have the identity $\mathcal{L}(w) = \big((b-1)n_c/N\big)\tilde{\mathcal{L}}_b(w) + \big((N-(b-1)n_c)/N\big)\Delta\mathcal{L}_b(w)$.

First, we observe that given the previously transmitted data samples, the gradient $\nabla \ell(w_b^{j-1}, \xi_b^j)$ is an unbiased estimate of the gradient $\nabla \tilde{\mathcal{L}}_b(w)$ of the empirical loss limited to the samples available at the edge node at block b. In formulas, $\mathrm{E}_{\xi_b^j|\tilde{\mathcal{X}}_b}[\nabla \ell(w,\xi_b^j)] = \nabla\tilde{\mathcal{L}}_b(w)$, where $\mathrm{E}_{\xi_b^j|\tilde{\mathcal{X}}_b}[\,\cdot\,]$ is the conditional expectation given the previously transmitted samples. We finally make the following assumption (see, e.g., [9]):

$(A4)$ For any set $\tilde{\mathcal{X}}_b$ of samples available at the edge node, there exist scalars $M \geq 0$ and $M_V \geq 0$ such that

$$\mathrm{V}_{\xi_b^j|\tilde{\mathcal{X}}_b}[\nabla \ell(w,\xi_b^j)] \leq M + M_V ||\nabla \tilde{\mathcal{L}}_b(w)||_2^2 \tag{9}$$

where $\mathrm{V}[\,\cdot\,] = \mathrm{E}[||\cdot||^2] - ||\mathrm{E}[\,\cdot\,]||^2$ is the variance.

4 Convergence analysis

In this section, we present our main result and its implications on the optimal choice of the number n_c of transmitted samples per block. Henceforth, we use the notation $\mathrm{E}_b[\,\cdot\,]$ to indicate the conditional expectation $\mathrm{E}_{\xi_b^1,\ldots\xi_b^{n_p}|\tilde{\mathcal{X}}_b}[\,\cdot\,]$ on the samples selected for the SGD updates in the b-th block given the set $\tilde{\mathcal{X}}_b$ of samples available at the edge node at b. We similarly define $\mathrm{E}_{B_l}[\,\cdot\,] = \mathrm{E}_{\xi_{B_l}^1,\ldots,\xi_{B_l}^{n_l}}[\,\cdot\,]$ as the conditional expectation on the samples selected for the SGD updates in block B_l (see Fig. 2(b)).

Theorem 1 *Under assumptions $(A1)$-$(A4)$, assume that the SGD stepsize α satisfies*

$$0 < \alpha \leq \frac{2}{LM_G} \tag{10}$$

and define

$$\gamma = \alpha\Big(1 - \frac{1}{2}\alpha L M_G\Big). \tag{11}$$

Then, for any sequence $\tilde{\mathcal{X}}_1, \ldots, \tilde{\mathcal{X}}_B$ the expected optimality gap at time T is upper bounded as

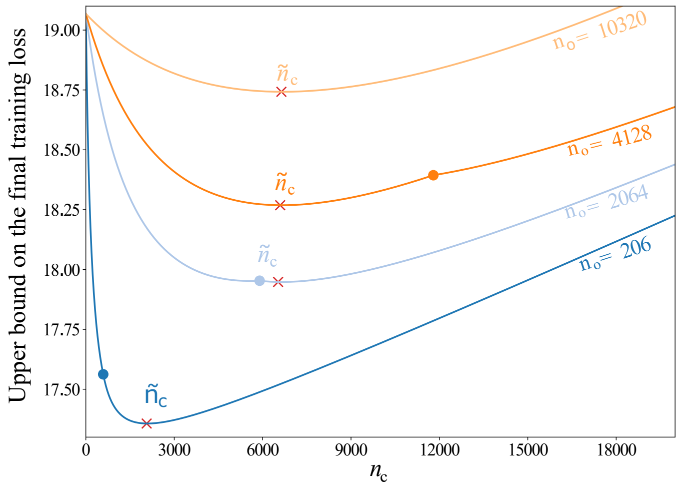

Figure 3: Upper bound (14)-(15) versus block size n_c for various values of the overhead n_o. The full dots represent values of n_c at which we have $T = B_d(n_c + n_o)$ (see Fig. 2), crosses represent the optimized value $\tilde{n}_c$.

$$
\begin{aligned}
&\mathrm{E}_B[\mathcal{L}(w_B^{n_p}) - \mathcal{L}(w^*)] \\
&\leq \frac{\alpha^2 LM}{2\gamma c}\frac{(B-1)}{B_d} + \Big(1 - \frac{(B-1)}{B_d}\Big)\mathrm{E}_B\Big[\Delta\mathcal{L}_B(w_B^{n_p}) - \Delta\mathcal{L}_B(w^*)\Big] \\
&+ \frac{1}{B_d}\sum_{l=1}^{B-1}(1-\gamma c)^{ln_p}\mathrm{E}_{B-l}\Big[\mathcal{L}_{B-l}(w_{B-l}^{n_p}) - \mathcal{L}_{B-l}(w^*) - \frac{\alpha^2 LM}{2\gamma c}\Big]
\end{aligned} \tag{12}
$$

if $T \leq B_d(n_c + n_o)$; *and by*

$$
\begin{aligned}
&\mathrm{E}_{B_l}\Big[\mathcal{L}\Big(w_{B_l}^{n_l}\Big) - \mathcal{L}(w^*)\Big] \leq \frac{\alpha^2 LM}{2\gamma c} \\
&+ \frac{1}{B_d}(1-\gamma c)^{n_l}\sum_{l=0}^{B_d-1}(1-\gamma c)^{ln_p}\mathrm{E}_{B_d-l}\Big[\mathcal{L}_{B_d-l}(w_{B_d-l}^{n_p}) - \mathcal{L}_{B_d-l}(w^*) - \frac{\alpha^2 LM}{2\gamma c}\Big]
\end{aligned} \tag{13}
$$

if $T > B_d(n_c + n_o)$.
Proof: See Appendix A.

The bound (12)-(13) extends the classical analysis of the convergence of SGD for the case in which the entire dataset is available at the learner [9, Theorem 4.6] to the set up under study. The bound distinguishes the case in which the edge node has the entire data set by the last block, and the complementary case, as seen in Fig. 2.

The first term in the bound (13) represents an asymptotic bias that does not vanish with the number of SGD updates, even when all the data points are available at the edge node. It is due to the variance (9) of the stochastic gradient. The bound (12) for smaller values of T also comprises an additional bias term, that is the second term in (13), due to the lack of knowledge about samples not received at the edge node by the end of the training process. In contrast, the last term in bound (12)-(13) accounts for the standard geometric decrease of the initial error in gradient-based learning algorithms. Here, the initial error for each block b is given by $\mathrm{E}_b\big[\mathcal{L}(w_{b-1}^{n_p}) - \mathcal{L}(w^*)\big]$. Note that the additional factor with exponent n_l in (13) accounts for the number of updates made after all the samples have been received at the edge node.

The bound (12)-(13) can be in principle optimized numerically in order to find an optimal value to the block size n_c. However, in practice, doing so would require fixing the choice of the sequence $\tilde{\mathcal{X}}_1, \ldots, \tilde{\mathcal{X}}_B$, and running Monte Carlo experiments for every randomly selected sample of the sequence of SGD updates (2), which is computationally intractable. Therefore, in the following, we derive a generally looser bound that can be directly evaluated numerically without running any Monte Carlo simulations. This bound will then be used in order to obtain an optimized value for n_c.

Corollary 1 *Under the conditions of Theorem 1, the expected optimality gap at time T is upper bounded as*

$$\mathrm{E}_B[\mathcal{L}(w_B^{n_p}) - \mathcal{L}(w^*)] \leq \frac{\alpha^2 LM}{2\gamma c}\frac{(B-1)}{B_d} + \Big(1 - \frac{(B-1)}{B_d}\Big)\frac{LD^2}{2}$$

$$+ \frac{1}{B_d}\sum_{l=1}^{B-1}(1-\gamma c)^{ln_p}\Big[\frac{LD^2}{2} - \frac{\alpha^2 LM}{2\gamma c}\Big], \tag{14}$$

if $T \leq B_d(n_c + n_o)$; and by

$$\mathrm{E}_{B_l}[\mathcal{L}(w_{B_l}^{n_l}) - \mathcal{L}(w^*)] \leq \frac{\alpha^2 LM}{2\gamma c} + \frac{1}{B_d}(1-\gamma c)^{n_l}\sum_{l=0}^{B_d-1}(1-\gamma c)^{ln_p}\Big[\frac{LD^2}{2} - \frac{\alpha^2 LM}{2\gamma c}\Big] \tag{15}$$

if $T > B_d(n_c + n_o)$.
Proof: See Appendix B.

We plot bound (14)-(15) in Fig. 3. These results are obtained for $N = 18,576$, $T = 1.5N$, $L = 1.908$, $c = 0.061$, $M = 1$, $M_G = 1$, $\tau_p = 1$, $\alpha = 0.0001$. We note that L and c represent respectively the smallest and largest eigenvalues of the data Gramian matrix for the example studied in Sec. 5. For each value of n_o, we mark in the figure both the value of n_c that minimizes the upper bound in Corollary 1 and the value of n_c at which we have the condition $T = B_d(n_c + n_o)$. As seen in Fig. 2, this is the minimum value of n_c that allows the full transmission of the training set by the last training block.

A first observation is that the optimized value of n_c, henceforth referred to as $\tilde{n}_c$, is generally smaller than the number N of training points in $\mathcal{X}$, suggesting the advantages of pipelining communication and computation. Furthermore, as the overhead n_o increases, it becomes preferable, in terms of the bound (14)-(15), to choose larger values $\tilde{n}_c$ for the block size n_c. This is because a larger value of n_o needs to be amortized by transmitting more data in each block, lest the transmission time is dominated by overhead transmission. Finally, for smaller values of n_o, the minimum $\tilde{n}_c$ of the bound is obtained when the entire data set is eventually transferred to the edge node, i.e., $T > B_d(n_c + n_o)$, while the opposite is true for larger value of n_o. Interestingly, this suggests that it may be advantageous in terms of final training loss, to forego the transmission of some training points in exchange for more time to carry out training on a fraction of the data set.

5 Numerical experiments

In this section, we validate the theoretical findings of the previous sections by means of a numerical example based on ridge regression on the California Housing dataset [11]. The dataset contains 20640 covariate vectors $x_n \in \mathrm{R}^8$, each with a real label y_n. We randomly select 90% of the samples to define the set $\mathcal{X}$ for training, i.e., we have $N = 18576$. As for Fig. 4, we choose $\tau_p = 1$ and $\alpha = 0.0001$. The parameter vector is initialized using i.i.d. zero-mean Gaussian entries with unitary power. The loss function is defined as $\ell(w,x) = (w^T x - y)^2 + \frac{\lambda}{N}||w||^2$ where $w \in \mathrm{R}^8$ and the regularization coefficient is chosen as $\lambda = 0.05$.

By computing the average final training loss for each value of n_c, we can experimentally determine the optimal value n_c^* of the block size. We compare the performance using this experimental optimum with the performance obtained using the minimum $\tilde{n}_c$ of the bound (14)-(15). To this end, in Fig. 4, given a fixed overhead size n_o, we plot the average training loss $\mathcal{L}(w_b^j)$ against the normalized training time j for n_c^* and for the value $\tilde{n}_c$ obtained from the bound (14)-(15). As references, we also plot as dotted lines the losses obtained for selected values of n_c. The choice of the block size n_c minimizing the average final loss is seen to be a trade-off between the rate of decrease of the loss and the final attained accuracy. In particular, decreasing n_c allows the edge node to reduce the loss more quickly, albeit with noisier updates and at the cost of a potentially larger final training loss due to the transmitted packet being dominated by the overhead. Importantly, determining the optimum block size experimentally instead of using bound (14)-(15) only provides a gain of 3.8% in terms of the final training loss, at the cost of a computationally burdensome parameter optimization.

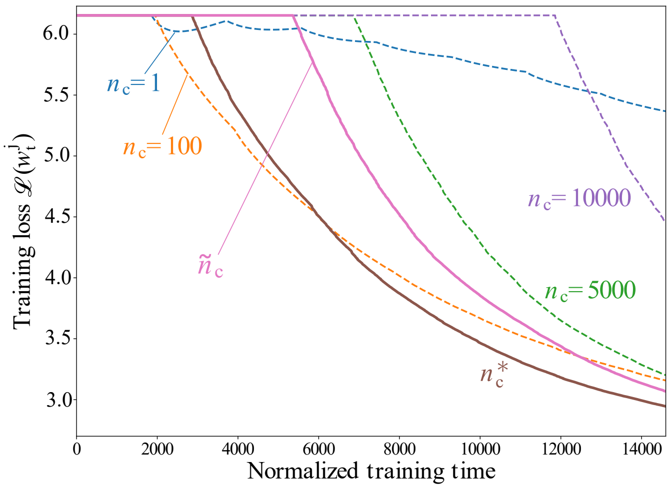

Figure 4: Training loss versus training time for different values of the block size n_c. Solid line: experimental and theoretical optima.

6 Conclusions

In this work, we considered an edge computing system in which an edge learner carries out training over a limited time period while receiving the training data from a device through a communication link. Considering a strategy that allows communication and computation to be pipelined, we have analysed the optimal communication block size as a function of the packet overhead. Among interesting directions for future work, we mention the inclusion of the effect of delays due to errors in the communication channel. In this case, the optimization problem could be generalized to account for the selection of the data rate. Other interesting extensions would be to consider online learning, where data sent in previous packets can be only partially stored at the server, and to investigate a scenario with multiple devices.

A Proof of Theorem 1

Using the same arguments as in the proof of [9, Theorem 4.6], we can directly obtain the following inequality for each block b:

$$\begin{aligned}&\mathrm{E}_b[\tilde{\mathcal{L}}_b(w_b^{n_p}) - \tilde{\mathcal{L}}_b(w^*)]\\&\leq \frac{\alpha^2 LM}{2\gamma c} + (1-\gamma c)^{n_p}\mathrm{E}_b\Big[\tilde{\mathcal{L}}_b(w_b^0) - \tilde{\mathcal{L}}_b(w^*) - \frac{\alpha^2 LM}{2\gamma c}\Big].\end{aligned} \tag{16}$$

Note that we have $w_b^0 = w_{b-1}^{n_p}$, since the initial parameter at block b is the final parameter obtained at block $b-1$. By definition of the local empirical losses (6)-(7), we have the equality

$$\tilde{\mathcal{L}}_b(w_{b-1}^{n_p}) = \frac{b-2}{b-1}\tilde{\mathcal{L}}_{b-1}(w_{b-1}^{n_p}) + \frac{1}{b-1}\mathcal{L}_{b-1}(w_{b-1}^{n_p}). \tag{17}$$

Plugging (17) into (16), we have

$$\begin{aligned}&\mathrm{E}_b[\tilde{\mathcal{L}}_b(w_b^{n_p}) - \tilde{\mathcal{L}}_b(w^*)]\\&\leq \frac{\alpha^2 LM}{2\gamma c} + (1-\gamma c)^{n_p}\mathrm{E}_b\Big[\Big(\frac{b-2}{b-1}\Big)\Big(\tilde{\mathcal{L}}_{b-1}(w_{b-1}^{n_p}) - \tilde{\mathcal{L}}_{b-1}(w^*)\Big)\\&+ \frac{1}{b-1}\Big(\mathcal{L}_{b-1}(w_{b-1}^{n_p}) - \mathcal{L}_{b-1}(w^*)\Big) - \frac{\alpha^2 LM}{2\gamma c}\Big].\end{aligned} \tag{18}$$

Iterating this substitution for all blocks $b-1, b-2, \ldots, 2$, we obtain

$$\begin{aligned}&\mathrm{E}_b[\tilde{\mathcal{L}}_b(w_b^{n_p}) - \tilde{\mathcal{L}}_b(w^*)] \leq \frac{\alpha^2 LM}{2\gamma c}\\&+ \sum_{l=1}^{b-1}(1-\gamma c)^{l n_p}\frac{1}{b-1}\mathrm{E}_b\Big[\mathcal{L}_{b-l}(w_{b-l}^{n_p}) - \mathcal{L}_{b-l}(w^*) - \frac{\alpha^2 LM}{2\gamma c}\Big].\end{aligned} \tag{19}$$

While inequality (19) applies for any choice of T, we now specialize the result to the case where the allocated amount of time T is not sufficient to transmit the whole dataset, i.e., $T \leq B_d(n_c + n_o)$. (see Fig. 2(a)). According to (6)-(8), for this case, we have the equality

$$\mathcal{L}(w) = \frac{(b-1)}{B_d}\tilde{\mathcal{L}}_b(w) + \frac{N-(b-1)}{B_d}\Delta\mathcal{L}_b(w). \tag{20}$$

Plugging (20) into (19) for block $b = B$, we then obtain

$$\begin{aligned}&\mathrm{E}_B[\mathcal{L}(w_B^{n_p}) - \mathcal{L}(w^*)]\\&\leq \frac{\alpha^2 LM}{2\gamma c}\frac{(B-1)}{B_d} + \Big(1 - \frac{(B-1)}{B_d}\Big)\mathrm{E}_b\Big[\Delta\mathcal{L}_B(w_B^{n_p}) - \Delta\mathcal{L}_B(w^*)\Big]\\&+ \frac{1}{B_d}\sum_{l=1}^{B-1}(1-\gamma c)^{l n_p}\mathrm{E}_B\Big[\mathcal{L}_{B-l}(w_{B-l}^{n_p}) - \mathcal{L}_{B-l}(w^*) - \frac{\alpha^2 LM}{2\gamma c}\Big],\end{aligned} \tag{21}$$

which is (12) in Theorem 1.

Finally, we consider the case where there is sufficient time to transmit the whole dataset, i.e., $T > B_d(n_c + n_o)$ (see Fig. 2(b)). According to (16), we have

$$\begin{aligned}&\mathrm{E}_{B_l}[\mathcal{L}_{B_l}(w_{B_l}^{n_l}) - \mathcal{L}_{B_l}(w^*)]\\&\leq \frac{\alpha^2 LM}{2\gamma c} + (1-\gamma c)^{n_l}\mathrm{E}_{B_l}\Big[\mathcal{L}(w_{B_l}^0) - \mathcal{L}(w^*) - \frac{\alpha^2 LM}{2\gamma c}\Big]\\&\overset{(a)}{\leq} \frac{\alpha^2 LM}{2\gamma c} + \frac{1}{B_d}(1-\gamma c)^{n_l}\sum_{l=0}^{B_d-1}(1-\gamma c)^{l n_p}\\&\cdot\ \mathrm{E}_{B_l}\Big[\mathcal{L}_{B_d-l}(w_{B_d-l}^{n_p}) - \mathcal{L}_{B_d-l}(w^*) - \frac{\alpha^2 LM}{2\gamma c}\Big],\end{aligned} \tag{22}$$

where (a) arises from plugging (21) in (22) with $B = B_d$. This is (13) in Theorem 1, concluding the proof.

B Proof of Corollary 1

Defining for all $t = 1, \ldots, B_d$, the optimum solution $\Delta w_b^* = \arg\min_w \Delta\mathcal{L}_b(w)$, we can write $\Delta\mathcal{L}_b(\Delta w_b^*) \leq \Delta\mathcal{L}_b(w^*)$, and hence also the inequality

$$\Delta\mathcal{L}_b(w_b^{n_p}) - \Delta\mathcal{L}_b(w^*) \leq \Delta\mathcal{L}_b(w_b^{n_p}) - \Delta\mathcal{L}_b(\Delta w_b^*). \tag{23}$$

Writing the Lipschitz continuity property of the gradients $(A2)$ with $\nabla(\Delta\mathcal{L}_b(\Delta w_b^*)) = 0$ and $(A1)$, we have $\Delta\mathcal{L}_b(w_b^{n_p}) - \Delta\mathcal{L}_b(\Delta w_b^*) \leq \frac{LD^2}{2}$. Using a similar argument, we can write $\mathcal{L}_b(w_b^{n_p}) - \mathcal{L}_b(w_b^*) \leq \frac{LD^2}{2}$, where $w_b^* = \arg\min_w \mathcal{L}_b(w)$. Plugging this into (21), we obtain the inequality

$$\begin{aligned} \mathrm{E}_B[\mathcal{L}(w_B^{n_p}) - \mathcal{L}(w^*)] &\leq \frac{\alpha^2 LM}{2\gamma c}\frac{(B-1)}{B_d} \\ &+ \Big(1 - \frac{(B-1)}{B_d}\Big)\frac{LD^2}{2} + \frac{1}{B_d}\sum_{l=1}^{B-1}(1-\gamma c)^{ln_p}\Big[\frac{LD^2}{2} - \frac{\alpha^2 LM}{2\gamma c}\Big], \end{aligned} \tag{24}$$

which is (14) in Corollary 1. Following the same approach with (22), we obtain

$$\begin{aligned} \mathrm{E}_{B_l}[\mathcal{L}(w_{B_l}^{n_l}) - \mathcal{L}(w^*)] &\leq \frac{\alpha^2 LM}{2\gamma c} \\ &+ \frac{1}{B_d}(1-\gamma c)^{n_l}\sum_{l=0}^{B_d-1}(1-\gamma c)^{ln_p}\Big[\frac{LD^2}{2} - \frac{\alpha^2 LM}{2\gamma c}\Big], \end{aligned} \tag{25}$$

which is (15) in Corollary 1, completing the proof.

References

[1] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge." [Online]. Available: http://arxiv.org/abs/1812.02858

[2] D. S. N. S. Yun Chao Hu, Milan Patel and V. Young, *Mobile Edge Computing A key technology towars 5G*. Sophia Antipolis, France: ETSI (European Telecommunications Standards Institute, 2015.

[3] U. Mohammad and S. Sorour, "Adaptive task allocation for mobile edge learning." [Online]. Available: http://arxiv.org/abs/1811.03748

[4] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *IEEE INFOCOM 2018 Proc*, April 2018, pp. 63–71.

[5] S. Teerapittayanon, B. McDanel, and H. T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *2017 IEEE Conf on Distributed Computing Systems (ICDCS)*, June 2017.

[6] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 4424–4434.

[7] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging." [Online]. Available: http://arxiv.org/abs/1602.05629

[8] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air." [Online]. Available: http://arxiv.org/abs/1901.00844

[9] L. Bottou, F. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[10] O. Simeone, *A Brief Introduction to Machine Learning for Engineers*. F&T in Signal Processing, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8453245

[11] R. K. Pace and R. Barry, "Sparse spatial autoregressions," *Statistics and Probability Letters*, vol. 33, pp. 291–297, 1997.