Faster Federated Learning with Decaying Number of Local SGD Steps

Jed Mills, Jia Hu, Geyong Min

Abstract—In Federated Learning (FL) client devices connected over the internet collaboratively train a machine learning model without sharing their private data with a central server or with other clients. The seminal Federated Averaging (FedAvg) algorithm trains a single global model by performing rounds of local training on clients followed by model averaging. FedAvg can improve the communication-efficiency of training by performing more steps of Stochastic Gradient Descent (SGD) on clients in each round. However, client data in real-world FL is highly heterogeneous, which has been extensively shown to slow model convergence and harm final performance when $K > 1$ steps of SGD are performed on clients per round. In this work we propose decaying K as training progresses, which can jointly improve the final performance of the FL model whilst reducing the wall-clock time and the total computational cost of training compared to using a fixed K. We analyse the convergence of FedAvg with decaying K for strongly-convex objectives, providing novel insights into the convergence properties, and derive three theoretically-motivated decay schedules for K. We then perform thorough experiments on four benchmark FL datasets (FEMNIST, CIFAR100, Sentiment140, Shakespeare) to show the real-world benefit of our approaches in terms of real-world convergence time, computational cost, and generalisation performance.

Index Terms—Federated Learning, Deep Learning, Edge Computing, Computational Efficiency.

1 Introduction

Federated Learning (FL) is a recent distributed Machine Learning (ML) paradigm that aims to collaboratively train an ML model using data owned by clients, without those clients sharing their training data with a central server or other participating clients. Practical applications of FL range from 'cross-device' scenarios, with a huge number of unreliable clients each possessing a small number of samples, to 'cross-silo' scenarios with fewer, more reliable clients possessing more data [1]. FL has huge economic potential, with cross-device tasks including mobile-keyboard next-word prediction [2], voice detection [3], and even as proof-of-work for blockchain systems [4]. Cross-silo tasks include hospitals jointly training healthcare models [5] and financial institutions creating fraud detectors [6]. FL has been of particular interest for training large Deep Neural Networks (DNNs) due to their state-of-the-art performance across a wide range of tasks.

Despite FL's great potential for privacy-preserving ML, there exist significant challenges to address before FL can be more widely adopted at the network edge. These include:

- *Heterogeneous client data:* each client device generates its own data, and cannot share it with any other device. The data between clients is therefore highly heterogeneous, which has been shown theoretically and empirically to harm the convergence and final performance of the FL model.
- *High communication costs:* many FL algorithms operate in rounds that involve sending the FL model parameters between the clients and the coordinating server thousands of times. Considering the bandwidth constraints of wireless edge clients, communication represents a major hindrance to training.
- *High computation costs:* training ML models has a high computational cost (especially for modern DNNs with a huge number of parameters). FL clients are typically low-powered (often powered by battery), so computing the updates to the FL model is a substantial bottleneck.
- *Wireless edge constraints:* clients are connected to the network edge and can range from modern smartphones to Internet-of-Things (IoT) devices. They are highly unreliable and can leave and join the training process at any time.

To address some of the above challenges, McMahan *et al.* proposed the Federated Averaging (FedAvg) algorithm [7]. FedAvg is an iterative algorithm that works in communication rounds, where in each round clients download a copy of the 'global model' to be trained, perform K steps of Stochastic Gradient Descent (SGD) on their local data, then upload their models to the coordinating server, which averages them to produce the next round's global model. Therefore, FedAvg works similarly to distributed-SGD (dSGD) as used in the datacentre, but more than one gradient is calculated by clients per communication round. Using $K > 1$ local steps improves the per-round convergence rate compared to dSGD (hence saving on communication), and FedAvg only requires a fraction of all clients to participate in each round, mitigating the impact of unreliable clients and stragglers.

Increasing K improves the convergence rate (in terms of communication rounds) of FedAvg, however it has been demonstrated that it comes at the cost of harming the minimum training error and maximum validation accuracy than can be achieved, especially when client data is heterogeneous [1], and large values of K show diminishing returns for convergence speed. Therefore the total amount of computation performed to reach a given model error can be significantly greater compared to datacentre training,

- *J. Mills, J. Hu and G. Min are with the Department of Computer Science, University of Exeter, EX4 4QF, United Kingdom. E-mail: {jm729, j.hu, g.min}@exeter.ac.uk. Corresponding authors: Jia Hu, Geyong Min.*

leading to concerns over the energy cost of FL [8]. Furthermore, the computation time on low-powered FL clients is not negligible, so improving communication-efficiency by using larger K can lead to a long training procedure [9], [10].

The primary reason behind the performance degradation with increasing K is 'client-drift' [11]: as the data between clients is non-Independent and Identically Distributed (non-IID), the minimum point(s) of each client's objective will be different. During local training, client models diverge (drift) towards their disparate minimisers, and the average of these disparate models may not have good performance. The extent of client-drift has been shown theoretically to be proportional to the level of heterogeneity between client data, the client learning rate (η), and K [11], [12].

One theoretically-justified method of addressing the problem of client-drift is to reduce η during training. Intuitively, if the learning rate is smaller then client models can move less far apart during the local update. Previous works have shown that decaying η is required for the error of the global model to converge to 0. In this paper, we propose instead decaying K to achieve a similar goal. Decreasing K addresses client-drift whilst reducing the real-time and computational cost of each FedAvg round. We show in experiments using benchmark FL datasets that decaying K can match or outperform decaying η in terms of time to converge to a given error, total computational cost, and maximum validation accuracy achieved by the model. The main contributions of this paper are as follows:

- We analyse the convergence of FedAvg when using a decreasing value of K for strongly-convex objectives, which provides novel insight into the constraints on K and η, and intuitively demonstrates why and demonstrates the impact of $K > 1$ on convergence.
- We derive the optimal value of K for any point during the training runtime, and use this optimal value to propose two theoretically-motivated approaches for decaying K based either on the communication round or the relative FL model error. We also use the analysis to derive the optimal value of η for later comparison.
- We perform extensive experiments using four benchmark FL datasets (FEMNIST, CIFAR100, Sentiment140, Shakespeare) to show that the proposed decaying-K scheme can reduce the amount of real-time taken to achieve a given model error, as well as improving final model validation performance.
- We present a further practical heuristic for decaying K based on training error which also shows excellent performance in terms of improving the validation performance of the FL model on the four benchmark datasets.

The rest of this paper is organised as follows: in Section 2 we cover related works that analyse the convergence properties of FedAvg, algorithms designed to address client-drift, and relevant developments in datacentre-based training; in Section 3 we formalise the FL training objective, analyse the convergence of FedAvg using our proposed decaying-K schedule, derive the optimal value of K during training, and use this to motivate three K-decay schemes; in Section 4 we present an experimental evaluation of the proposed schemes; and in Section 5 we conclude the paper.

2 Related Work

In this section, we cover works that study the theoretical convergence properties of FedAvg (and related algorithms) and client-drift, algorithms that improve the convergence of FL, and works that study related problems in the datacentre setting.

2.1 Analysis of FedAvg

There has been significant research efforts in theoretically analysing the convergence of FedAvg. Li *et al.* [12] proved a convergence rate of $\mathcal{O}(1/T)$ (where T is equal to the number of total iterations, rather than communications rounds) on strongly-convex objectives. Their analysis suggests that an optimal number of local steps (K) exists to minimise the number of communication rounds to reach ϵ-precision, and the authors highlighted the need to decay the learning rate (η) during training. Non-dominant convergence in terms of total iterations remains an open problem within FL. Karimireddy *et al.* [11] added a server learning rate to FedAvg to prove its convergence for nonconvex objectives. Charles and Konecný [13] analysed the convergence of Local-SGD methods (including FedAvg) for quadratic objectives to gain insights into the trade-off between convergence rate and final model accuracy. Malinovsky *et al.* [14] generalised Local-SGD methods to generic fixed-point functions to analyse the effect of K on the ϵ-accuracy. Yang *et al.* [15] were the first to achieve linear speedup in terms of number of participating workers for FedAvg on nonconvex objectives. However, when considering partial worker participation (which is a key element of the FL scenario), their analysis does not show speedup with respect to K. Previous works have also analysed the convergence of FedAvg from perspectives such as minimising the total energy cost and optimal resource allocation [16].

While the above works analyse the convergence of FedAvg in terms of total iterations and/or communication rounds, the runtime of FedAvg is affected by multiple factors: model convergence rate, total number of local SGD steps, communication bandwidth, model size, and the compute power of client devices. These factors must all be considered if the objective is to improve the runtime of FedAvg, as we do in this work.

2.2 Novel FL Algorithms

Due to FL's long training times and the challenging distributed edge environment, a large number of novel algorithms have been designed to improve the convergence rate of FedAvg. Li *et al.* [17] proposed FedProx, which adds a proximal term to client objectives penalising the distance to the current global model. Karimireddy *et al.* [11] added Stochastic Variance-Reduced Gradients (SVRG) to FedAvg in SCAFFOLD, demonstrating significant speedup on popular FL benchmarks. Empirical convergence rates have also been improved by adding adaptive optimisation to FedAvg both locally [10] and globally [18]. Adaptive optimisation has also been implemented during the server-update of FedAvg [19], which can accelerate convergence without increasing the per-round communication or computation costs for clients. A recent survey covering developments in FL algorithms and their relation to the communications properties of FL is given in [20].

The above algorithms can be considered variants of FedAvg in that they perform rounds of local training and model averaging. Our proposed method of decaying the number of local steps during training could in principle be used with any FedAvg variant, which is a potential avenue for future research.

2.3 Datacentre Training

Distributed training in the datacentre shares similarities with the FL scenario, and there exists a substantial body of work studying datacentre training. The classic datacentre-based algorithm is distributed-SGD (dSGD), where nodes each compute a single (often very large) minibatch gradient and send it to the parameter server for aggregation. Woodworth *et al.* [21] proved for quadratic objectives that Local-SGD methods (which perform multiple steps of SGD between aggregations) converge at least as fast as dSGD (in terms of total iterations), but that Local-SGD does not dominate for more general convex problems. Similarly, Wang *et al.* [22] unified the analysis of various algorithms related to Local-SGD, covering different communication topologies and non-IID clients, achieving state-of-the-art rates for some settings. Lin *et al.* [23] presented a thorough empirical study showing that local-SGD methods generalise better than large-batch dSGD, motivating their approach of switching from dSGD to local-SGD during the later stages of training. Another approach to improve the generalisation performance of large-batch dSGD are 'extra-gradient' methods that compute gradient updates after a step of SGD before applying them to the global model [24].

While these works present methods that variously improve runtime or generalisation performance, their findings cannot be directly applied to FL. From a theoretical perspective, the primary differences are FL's highly non-IID clients and very low per-round participation rates (which can be as low as 0.1% [25]). FL client also have much lower communication bandwidth and computational power compared to datacentre compute nodes.

3 FedAvg with Decaying Local Steps

We now formally describe the FL optimisation problem, theoretically analyse the convergence of FedAvg with a decaying number of local SGD steps, and present theoretically-motivated schedules based upon the analysis.

3.1 Problem Setup

In FL there are a large number of clients that each possess a small number of local samples. The objective is to train model $\boldsymbol{x}$ to minimise the expected loss over all samples and over all clients, namely:

$$F(\boldsymbol{x}) = \sum_{c=1}^{C} p_c f_c(\boldsymbol{x}) = \sum_{c=1}^{C} p_c \left[\sum_{n=1}^{n_c} f(\boldsymbol{x}, \xi_{c,n}) \right], \qquad (1)$$

where C is the total number of FL clients, p_c is the fraction of all samples owned by client c (such that $\sum_{c=1}^{C} p_c = 1$), f is the loss function used on clients, and $\{\xi_{c,1}, \cdots, \xi_{c,n_c}\}$ represent the training samples owned by client c.

To minimise $F(\boldsymbol{x})$ in a communication-efficient manner, FedAvg (presented in Algorithm 1) performs multiple steps of SGD on each client between model averaging. FedAvg operates in communication rounds, where in each round r a subset of clients $\mathcal{C}_r$ download the current global model $\boldsymbol{x}_r$ (line 5), perform K_r steps of SGD on their local dataset (lines 6-8), and then upload their new models to the coordinating server (line 9). The server averages the received client models to produce the next round's global model (line 11).

FedAvg is typically described and analysed as selecting a subset of clients uniformly at random to participate in each communication round (line 3). However in real-world FL deployments, clients generally do not participate uniformly at random due to their behaviour, communication and compute capabilities. The non-uniform participation of FL clients has lead to the research direction of 'fair' FL [26].

Algorithm 1: Federated Averaging [7]

1 **input:** initial global model $\boldsymbol{x}_0$, learning rate schedule $\{\eta_r\}$, local steps schedule $\{K_r\}$
2 **for** round $r = 1$ **to** R **do**
3 select round clients $\mathcal{C}_r$
4 **for** client $c \in \mathcal{C}_r$ in parallel **do**
5 download global model $\boldsymbol{x}_r$
6 **for** local SGD step $k = 1$ **to** K_r **do**
7 $\boldsymbol{x}^c_{r,k} \leftarrow \boldsymbol{x}_r - \eta_r \nabla f(\boldsymbol{x}^c_k, \xi^c_k)$
8 **end**
9 upload local model $\boldsymbol{x}^c_{r,K_r}$ to server
10 **end**
11 update global model $\boldsymbol{x}_{r+1} \leftarrow \frac{1}{|C_r|} \sum_{c \in C_r} \boldsymbol{x}^c_{r,K_r}$
12 **end**

The updates to clients models within the FedAvg process can be viewed from the perspective of communication rounds (as shown in most FL works and presented in Algorithm 1), but can also be reformulated in terms of a continuous sequence of SGD steps on each client, with updates periodically being replaced by averaging. Suppose we reindex the client models from $\boldsymbol{x}^c_{r,k}$ to $\boldsymbol{x}^c_t$ where t is the *global iteration*, $t \in \{1, \cdots, T\}$. Note that $\sum_r \{K_r\} = T$. For a given FL client i and local SGD step t the update to the local model $\boldsymbol{x}^i_t$ can be given as:

$$\begin{aligned} \boldsymbol{y}^i_{t+1} &= \boldsymbol{x}^i_t - \eta_t \nabla f(\boldsymbol{x}^i_t, \xi^i_t), \\ \boldsymbol{x}^i_{t+1} &= \begin{cases} \sum_{c \in \mathcal{C}_t} p_c \boldsymbol{y}^c_{t+1} & \text{if } t \in \mathcal{I}, \\ \boldsymbol{y}^i_{t+1} & \text{otherwise}, \end{cases} \end{aligned} \qquad (2)$$

where $\mathcal{I}$ is the set of indexes denoting the iterations at which model communication occurs (which will be equal to the cumulative sum of $\{K_r\}$). This formulation states that clients not participating in the current round compute and then discard some local updates, which is not true in reality but makes analysis more amenable and is theoretically equivalent to FedAvg as presented in Algorithm 1. We define the average client model at any given iteration t using (2) as: $\bar{\boldsymbol{x}}_t = \sum_{c=1}^{C} p_c \boldsymbol{x}^t_c$.

3.2 Runtime Model of FedAvg

Inspecting Algorithm 1 shows that the nominal wall-clock time for each client c to complete a communication round r is:

$$W^c_r = \frac{|\boldsymbol{x}|}{D^c} + K_r \beta^c + \frac{|\boldsymbol{x}|}{U^c}, \qquad (3)$$

where $|\boldsymbol{x}|$ is the size of the FL model (in megabits), U^c and D^c are the upload and download bandwidth of client c in megabits per second, and β^c is the per-minibatch computation time of client c. The nominal time to complete a round for client c is therefore the sum of the download, local compute, and upload times. Furthermore, as FL clients are usually connected wirelessly at the network edge and geographically dispersed, we assume that U^c and D^c are independent of the total number of participating clients. For wireless connections, typically $D^c >> U^c$.

For a single round, the server must wait for the slowest client (straggler) to send its update. Therefore the time taken to complete a single round W_r is:

$$W_r = \max_{c \in \mathcal{C}_r} \{W_r^c\}. \tag{4}$$

To simplify the FedAvg runtime model, we assume that all clients have the same upload bandwidth, download bandwidth, and per-minibatch compute time. That is, $U^c = U$, $D^c = C$, and $\beta^c = \beta$, $\forall\ c$. Using these simplifications, the total runtime W for R communication rounds of FedAvg are:

$$W = \sum_{r=1}^{R} W_r = R\left(\frac{|\boldsymbol{x}|}{D} + \frac{|\boldsymbol{x}|}{U}\right) + \beta \sum_{r=1}^{R} K_r. \tag{5}$$

Previous works consider a fixed number of local steps during training: $K_r = K$, $\forall\ r$. There are extensive works showing that a larger K can lead to an increased convergence rate of the global model [7]. However, large K means that fewer communication rounds can be completed in a given timeframe. Previous works have shown that due to the low computational power of FL clients, the value of β can dominate the per-round runtime [9]. Therefore by decaying K_r during training, a balance between fast convergence and higher round-completion rate can be achieved, which is the primary focus of this work.

3.3 Convergence Analysis

We now present a convergence analysis of FedAvg using a decaying number of local steps K_r and constant learning rate η. We make the following assumptions, which are typical of within the theoretical analysis within FL.

Assumption 1: Client objective functions are L-smooth:

$$f_c(\boldsymbol{x}) \leq f_c(\boldsymbol{y}) + (\boldsymbol{x} - \boldsymbol{y})^\top \nabla f_c(\boldsymbol{y}) + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

As $F(\boldsymbol{x})$ is a convex combination of f_c, then it is also L-smooth.

Assumption 2: Client objective functions are μ-strongly convex:

$$f_c(\boldsymbol{x}) \geq f_c(\boldsymbol{y}) + (\boldsymbol{x} - \boldsymbol{y})^\top \nabla f_c(\boldsymbol{y}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2,$$

with minima $f_c^* = \min f_c$. As $F(\boldsymbol{x})$ is a convex combination of f_c, then it is also μ-strongly convex.

Assumption 3: For uniformly sampled data points ξ_k^c on client c, the variance of stochastic gradients on c are bounded by:

$$\mathbb{E}\|\nabla f_c(\boldsymbol{x}; \xi_k^c) - \nabla f_c(\boldsymbol{x})\|^2 \leq \sigma_c^2.$$

Due to analysing gradient descent on an L-smooth function, the magnitude of the gradient is naturally bounded by the distance between the first iterate $\boldsymbol{x}_1$ and the minimiser:

$$\|\nabla F(\boldsymbol{x})\|^2 \leq L^2\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2.$$

In our later analysis we define $G^2 = L^2\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|^2$ for convenience, i.e. the maximum norm of the gradient during training.

As per [12], we quantify the extent of non-IID client data with:

$$\Gamma = F^* - \sum_{c=1}^{C} p_c f_c^*,$$

where F^* is the minimum point of $F(\boldsymbol{x})$. $\Gamma \neq 0$ when the minimiser of the global objectives is not the same as the average minimiser of client objectives. $\Gamma = 0$ if the FL data is IID over the clients.

Assumption 2 states that our analysis considers strongly-convex objectives. Although FL is typically used to train large DNNs (with nonconvex objectives), strongly-convex models are widely-used, for example in Support Vector Machines. Furthermore, the starte-of-the-art in analysing FedAvg's convergence behaviour lags behind the empirical developments, with contemporary anlyses also making the convex assumption [12] [16]. The experimental evaluations in Section 4 consider one convex model (Sentiment 140) and three nonconvex DNNs. We leave it to future work to derive optimal K schedules for nonconvex objectives.

Theorem 1: *Let Assumptions 1-3 hold, and define $\kappa = {}^{L}/_{\mu}$. The expected minimum gradient norm of FedAvg using a monotonically decreasing number of local SGD steps K_r and fixed stepsize $\eta \leq {}^{1}/_{4L}$ after T total iterations is given by:*

$$\min_t\{\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_t)\|^2\right]\} \leq \frac{2\kappa(\kappa F(\bar{\boldsymbol{x}}_0) - F^*)}{\eta T} + \eta\kappa L\left[\sum_{c=1}^{C} p_c^2\sigma_c^2 + 6L\Gamma + \left(8 + \frac{4}{N}\right)G^2\frac{\sum_{r=1}^{R} K_r^3}{\sum_{r=1}^{R} K_r}\right]. \tag{6}$$

Proof: See Appendix A.2.

The above theorem provides some useful insights into the convergence properties of FedAvg when using multiple local steps. Some of these are detailed below.

Remark 1.1: relation to centralised SGD. With a fixed learning rate and decreasing K_r, Theorem 1 shows that FedAvg converges with $\mathcal{O}({}^{1}/_{T}) + \mathcal{O}(\eta)$. This result reflects the classical result of centralised SGD with a fixed learning rate (albeit with different constants due to non-IID clients and $K_r > 1$). Previous works have shown (like in centralised SGD) the requirement for η to be decayed to allow FedAvg to converge arbitrarily close to the global minima [11], [12]. However in this work we are interested in the runtime and computational savings available when decaying K_r, so do not feel the need to prove the already-covered decaying η result here.

Remark 1.2: benefit of $K > 1$. When using a decreasing η, previous analyses have shown that $K > 1$ acts to reduce the variance introduced by client stochastic gradients (the $\sum_{c=1}^{C} p_c^2\sigma_c^2$ term) [11], [12]. Dividing (6) by K_r (to achieve the convergence result in terms of total number of rounds) shows the same benefit in our analysis. We also observe empirically that $K_r > 1$ helps to reduce the variance of the global model updates even with a fixed η. Similarly a large number of clients participating per round (N) helps to reduce the variance that is introduced by performing $K_r > 1$ steps. FedAvg deployments can therefore benefit more from sampling a larger number of clients per round N when the number of local steps K_r is large.

Remark 1.3: drawback of $K > 1$. The second term of Theorem 1 shows that using $K_r > 1$ harms the convergence of FedAvg in terms of total number of iterations T. This is the case for all state-of-the-art analyses save for quadratic objectives [21]. However, in FL we wish to minimise the number of communication rounds (due to the quantity of communicated data and impact of stragglers etc.) alongside the total number of iterations (both of which affect the runtime of FedAvg).

Remark 1.4: real-world participation rates. Our formulation of FedAvg our analysis assumes a constant participation rate, but in real-world FL the round participation rate varies [25]. Setting K_r to a large value makes more progress in a round, but fewer clients will be able to complete the round in a given timeframe. This poses an interesting trade-off between K_r and N, which could be a potential avenue for future research.

3.4 Optimal Values of K_r and η_r

FedAvg is an iterative algorithm with each round starting from a new global model. Therefore, each iteration can be viewed as restarting the algorithm, using model $\boldsymbol{x}_{t_0}, \forall t_0 \in \mathcal{I}$. Using this formulation, we can indepedently derive what the optimal fixed value of K would be at the start of any communication round during training. As training progresses and new rounds of training are completed, this value of K can therefore vary. When using a fixed number of local steps $K_r = K$ and communication rounds R, the total number of FedAvg iterations is $T = KR$. Substituting this into (5) gives the total runtime W of T iterations of FedAvg:

$$W = \frac{T}{K}\left[\frac{|\boldsymbol{x}|}{D} + \frac{|\boldsymbol{x}|}{U} + \beta K\right]. \tag{7}$$

Setting $K_r = K$ and substituting (7) into Theorem 1 gives us the convergence of t rounds of FedAvg for fixed K and η in terms of the runtime, starting from an arbitrary round in the training process $\boldsymbol{x}_{t_0}, \forall t \in \mathcal{I}$, rather than the number of iterations:

$$\begin{aligned}\min_{t>t_0}\{\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_t)\|^2\right]\} &\leq \frac{2\kappa(\kappa F(\bar{\boldsymbol{x}}_{t_0}) - F^*)}{\eta W K}\left[\frac{|\boldsymbol{x}|}{D} + \frac{|\boldsymbol{x}|}{U} + \beta K\right] \\ &+ \eta\kappa L\left[\sum_{c=1}^{C} p_c^2\sigma_c^2 + 6L\Gamma + \left(8 + \frac{4}{N}\right)G^2K^2\right]. \end{aligned} \tag{8}$$

As (8) gives the convergence of t rounds of FedAvg, starting from arbitrary point $\boldsymbol{x}_{t_0}$, with a fixed K and η. If the round index is instead substituted with a time index (with x_w corresponding to the value of x_t at time w), (8) can be used to determine what the optimal fixed valued of K looking forward would be for any point in time during the training process, K_w^*.

Theorem 2: *Let Assumptions 1-3 hold and define $\kappa = L/\mu$. For fixed $\eta \leq 1/4L$, the optimal number of local SGD steps K to minimise (8) is given by:*

$$K_w^* = \sqrt[3]{\frac{(\kappa F(\bar{\boldsymbol{x}}_{t_0}) - F^*)}{8\eta^2 L\,(1 + 1/2N)}\frac{(|\boldsymbol{x}|/D + |\boldsymbol{x}|/U)}{W}}. \tag{9}$$

Proof: See Appendix A.3.

Theorem 2 shows that K_w^* decreases at least as fast as $\mathcal{O}(1/\sqrt[3]{W})$, motivating the principal of decreasing the number of local steps during FedAvg. The decreasing value of the global model objective $F(\bar{\boldsymbol{x}}_{t_0})$ also influences K_w^*. The implications of Theorem 2 are discussed in the following Remarks.

Remark 2.1: relation to other works. Wang and Joshi [27] investigated variable communication intervals for the Periodic Averaging SGD (PASGD) algorithm in the datacentre, and found that the optimal interval decreased with $\mathcal{O}(1/\sqrt[3]{W})$. K_w^* decreases slower in FedAvg due to the looser bound on client divergence between averaging (scaling with K^2 rather than K).

Remark 2.2: dependence on client participation rate. As the number of clients participating per round (N) increases, K_w^* increases. This is because a higher number of participating clients decreases the variance in model updates (which is especially significant considering the non-IID client data).

Remark 2.3: reformulation using communication rounds. In FL, it is typically assumed that the local computation time is dominated by the communication time due to the low-bandwidth connections to the coordinating server. If we consider the case where ($|\boldsymbol{x}|/D + |\boldsymbol{x}|/U >> \beta K$), then $W \approx R\left(|\boldsymbol{x}|/D + |\boldsymbol{x}|/U\right)$. This means:

$$\begin{aligned} K_r^* &= \sqrt[3]{\frac{\kappa F(\bar{\boldsymbol{x}}_{t_0}) - F^*}{8\eta^2 L\,(1 + 1/2N)}\frac{1}{R}} \\ &\leq \sqrt[3]{\frac{\kappa F(\bar{\boldsymbol{x}}_{0}) - F^*}{8\eta^2 L\,(1 + 1/2N)}\frac{1}{R}}, \end{aligned} \tag{10}$$

where the inequality comes from the fact that $F(\boldsymbol{x}_t) \leq F(\boldsymbol{x}_0)$ given Assumption 1 and an appropraitely chosen stepsize η. K_r^* is not dependent on the local computation time, only the total number of rounds R. Using (10) as a decay scheme produces a fairly aggressive decay rate, and is tested experimentally in Section 4 using a variety of model types (which have different communication and computation times).

A similar approach can be taken to find the optimal value of η_r^* at each communication round. Although the focus of this paper is on decaying K to improve the convergence speed of FL, we compare it to the effect of decaying η as well as constant η and K.

Corollary 2.1: *Let Assumptions 1-3 hold and define $\kappa = L/\mu$. Given stepsizes $\eta_r \leq 1/4L$, the optimal value of η at any point in time during training to minimise (8) is given by:*

$$\eta_w^* = \sqrt{\frac{2(\kappa F(\bar{\boldsymbol{x}}_{t_0}) - F^*)}{LZ}\frac{(|\boldsymbol{x}|/D + |\boldsymbol{x}|/U + \beta K)}{W}},$$
$$\text{where } Z = \sum_{c=1}^{C} p_c^2\sigma_c^2 + 6L\Gamma + (8 + 4/N)G^2K^2. \tag{11}$$

Proof: See Appendix A.4.

Corollary 2.1 shows that the optimal value of η decreases with $\mathcal{O}(1/\sqrt{W})$. Several insights from Corollary 2.1 are given below.

Remark 2.1.1: impact of round time. (11) shows that η_w^* is directly affected by the per-round time: as any of the upload, download or computation time increases, η_w^* increases. This is because less progress is made over time (due to longer rounds) so

TABLE 1: Datasets and models used in the experimental evaluation (DNN = Deep Neural Network, CNN = Convolutional Neural Network, GRU = Gated Recurrent Network). K_0 and η_0 are the initial number of local steps and initial learning rate used.

Task	Type	Classes	Model	Model Size (Mb)	Total Clients	Clients per Round	Samples per Client	K_0	η_0
Sent140	Sentiment analysis	2	Linear	0.32	21876	50	15	60	3.0
FEMNIST	Image classification	62	DNN	6.71	3000	60	170	80	0.3
CIFAR100	Image classification	100	CNN	40.0	500	25	100	50	0.01
Shakespeare	Character prediction	79	GRU	5.21	660	10	5573	80	0.1

a larger $\eta^* + W$ compensates by making more progress per SGD step.

Remark 2.1.2: dependence on client participation rate. Similar to K_w^*, a larger number of clients participating per round (N) allows for a smaller η_w^* by reducing variance due to client-drift. Larger K in (11) also allows for smaller η as more progress is made per round.

Remark 2.1.3: reformulation using communication rounds. Making the same assumption as in (10) ($|\boldsymbol{x}|/_D + |\boldsymbol{x}|/_U >> \beta K$) gives a decay schedule for η_r in terms of the number of communication rounds:

$$\eta_r^* = \sqrt{\frac{2(\kappa F(\boldsymbol{x}_{t_0}) - F^*)}{LZ}\frac{1}{R}} \leq \sqrt{\frac{2(\kappa F(\boldsymbol{x}_0) - F^*)}{LZ}\frac{1}{R}}, \tag{12}$$

where Z is defined in (11), and the inequality again comes from using $F(\boldsymbol{x}_t) \leq F(\boldsymbol{x}_0)$. This decay schedule is also tested empirically in Section 4.

3.5 Schedules Based on Training Progress

In practice the values of κ, F^*, and L are difficult or impossible to evaluate due to complex nonlinear models (i.e. DNNs) and data privacy in FL. Therefore, appropriate values of K and η are chosen via grid-search or some other method (such as Bayesian Optimisation). Denote K_0 as a 'good' value of K at $W = 0$ (found via grid search), and K_r as the value of K to be used for round r. Each successive round of FedAvg can be considered as a new optimisation procedure with starting model $\bar{\boldsymbol{x}}_r$. If we make the further assumption that $F^* = 0$, substituting these two sets of values into (9) and dividing gives us K_r in terms of K_0:

$$K_r^* = \left\lceil \sqrt[3]{\frac{F(\bar{\boldsymbol{x}}_r)}{F(\bar{\boldsymbol{x}}_0)}}\, K_0 \right\rceil. \tag{13}$$

A similar process can be applied to find η_r in terms of η_0:

$$\eta_r^* = \sqrt[2]{\frac{F(\bar{\boldsymbol{x}}_r)}{F(\bar{\boldsymbol{x}}_0)}}\, \eta_0. \tag{14}$$

$F(\bar{\boldsymbol{x}}_r)$ is the training loss of the global model at the start of round r. Practically, an estimate of $F(\bar{\boldsymbol{x}}_r)$ can be obtained by requiring clients $c \in \mathcal{C}_r$ to send their training loss after the first step of local SGD to the server each round: $f_c(\bar{\boldsymbol{x}}_r, \xi_{c,0})$, where $\mathbb{E}\left[f_c(\bar{\boldsymbol{x}}_r, \xi_{c,0})\right] = F(\bar{\boldsymbol{x}}_r)$. This is only a single floating-point value that does not require any extra computation and negligibly increases the per-round communication costs.

Due to only a small fraction of the non-IID clients being sampled per round, the per-round variance of $\frac{1}{N}\sum_{c\in\mathcal{C}_r} f_c(\bar{\boldsymbol{x}}_r, \xi_{c,0})$ can be very high. Therefore, we propose a simple rolling-average estimate using window size s:

$$F(\bar{\boldsymbol{x}}_r) \approx \frac{1}{sN}\sum_{i=r-s}^{r}\sum_{c\in\mathcal{C}_i} f_c(\bar{\boldsymbol{x}}_i, \xi_{c,0}). \tag{15}$$

Our experiments in Section 4 use a window size $s = 100$, where our experiments run for at least $R = 10,000$ communication rounds. For the first s rounds when (15) cannot be computed, we keep $K_r = K_0$.

When using a fixed value of K, Theorem 1 shows that the minimum gradient norm converges with $\mathcal{O}(1/T) + \mathcal{O}(\eta K^2)$. As noted earlier, this result is analogous to the classical result of dSGD using a fixed learning rate. In the datacentre, the practical heuristic of decaying the learning rate η when the validation error plateaus is commonly used to allow the model to reach a lower validation error. We can therefore use a similar strategy for FedAvg: once the validation error plateaus we decay either K or η. We investigate this heuristic alongside the decay schedules presented above in Section 4.

4 Experimental Evaluation

In this section, we present the results of simulations comparing the three decaying-K schemes proposed in Section 3 to evaluate their benefits in terms of runtime, communicated data and computational cost on four benchmark FL datasets. Code to reproduce the experiments is available from: *github.com/JedMills/Faster-FL*.

4.1 Datasets and Models

To show the broad applicability of our approach, we conduct experiments on 4 benchmark FL learning tasks from 3 Machine Learning domains (sentiment analysis, image classification, sequence prediction) using 4 different model types (simple linear, DNN, Convolutional, Recurrent).

Sentiment 140: a sentiment analysis task of Tweets from a large number of Twitter users [28]. We limited this dataset to users with ≥ 10 samples, leaving 22k total clients, with 336k training and 95k validation samples, with an average of 15 training samples per client. We generated a normalised bag-of-words vector of size 5k for each sample using the 5k most frequent tokens in the dataset. We train a binary linear classifier (i.e. a convex model) using 50 clients per round (0.2% of all clients) and a batch size of 8.

FEMNIST: an image classification task of (28×28) pixel greyscale (flattened) images of handwritten letters and numbers from 62 classes, grouped by the writer of the symbol [28]. We

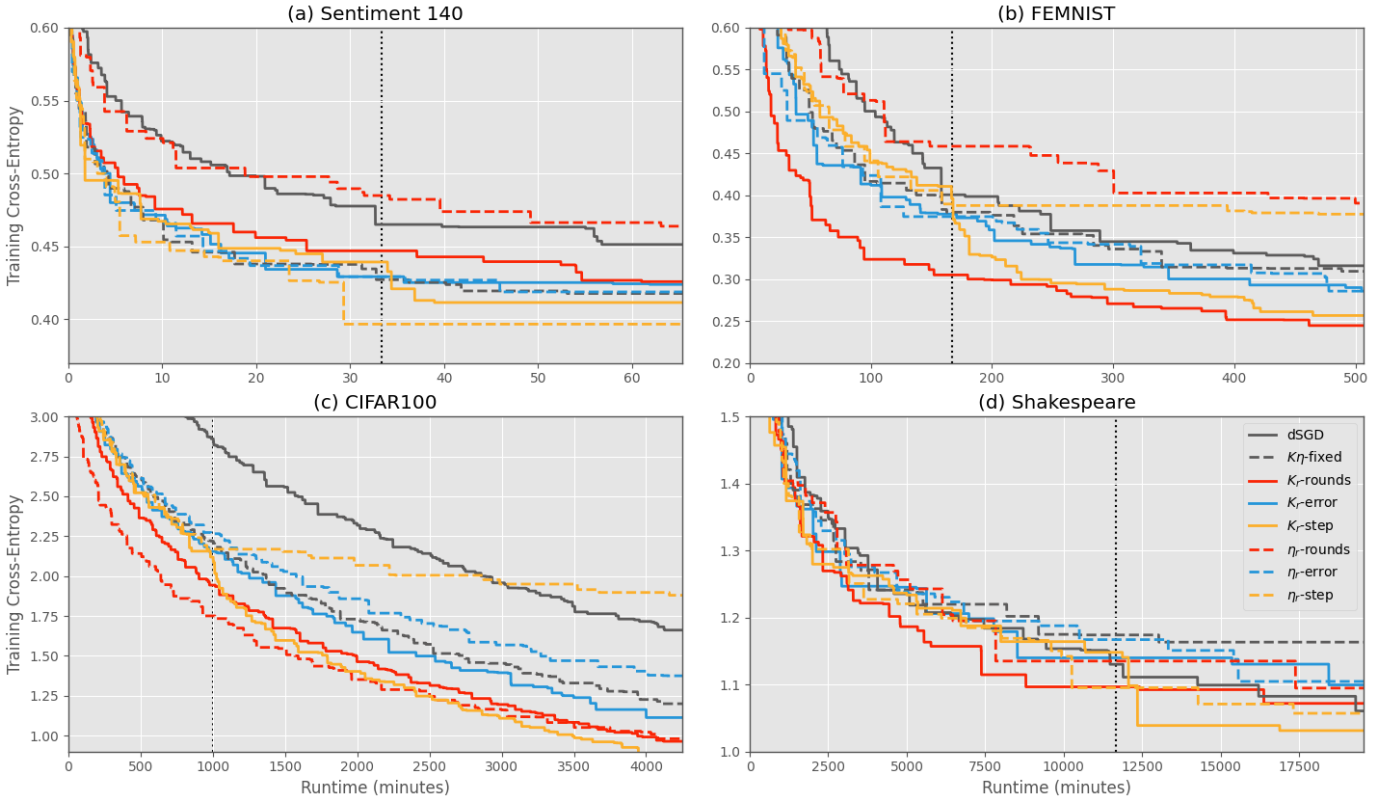

Fig. 1: Cumulative lowest training cross-entropy error over time of FedAvg using different schedules for K_r and η_r. Curves show mean over 5 random trials. Vertical line shows the communication round where the validation error plateaus.

used 3k total clients, with a total of 501k training and 129k validation samples, with an average of 170 training samples per client. We sample 60 clients per round (2% of all clients) with a batch size of 32. We train a DNN comprising a 200-unit ReLU Fully-Connected layer (FC), a second 200-unit ReLU FC layer, and a softmax output layer.

CIFAR100: an image classification task of (32×32) pixel RGB images of objects from 100 classes. We use the non-IID partition first proposed in [19], which splits the images into 500 clients based on the class labels. There are 50k training and 10k validation samples in the dataset, with each client possessing 100 samples. We select 25 clients per round (5% of all clients). We train a Convolutional Neural Network (CNN) consisting of two (3×3) ReLU convolutional + (2×2) Max-Pooling blocks, a 512-unit ReLU FC layer, and a softmax output layer. As per other FL works [19], [18] we apply random preprocessing composed of a random horizontal flip and crop of the (28×28) pixel sub-image to improve generalisation.

Shakespeare: a next-character prediction task using the complete plays of William Shakespeare [28]. The lines from all plays are partitioned by the speaking part in each play, and clients with ≤ 2 lines are discarded, leaving 660 total clients. Using a sequence length of 80, there are 3.7m training and 357k validation samples, with an average of 5573 training samples per client. We sample 10 clients per round (1.5% of all clients) with a batch size of 32. We train a Gated Recurrent Unit (GRU) DNN comprising a $79 \rightarrow 8$ embedding, two stacked GRUs of 128 units, and a softmax output layer.

4.2 Simulating Communication and Computation

The convergence of FedAvg for the learning tasks was simulated using Pytorch on GPU-equipped workstations. However, real-world FL runs distributed training on low-powered edge clients (such as smartphones and IoT devices). These clients exhibit much lower computational power and lower bandwidth to the coordinating server compared to datacentre nodes.

To realistically simulate real-world FedAvg, we use the runtime model presented in Section 3.2 and Equation (5). We assume that each client has a download bandwidth of $D = 20$ Mbps and an upload bandwidth of $U = 5$ Mbps. These are typical values for wireless devices connected via 4G LTE in the United Kingdom [29]. To determine the runtime of a minibatch of SGD on a typical low-powered edge device (β), we ran 100 steps of SGD for each learning task on a Raspberry Pi 3B+ with the following configuration:

- 1.4GHz 64-bit quad-core Cortex-A53 processor.
- 1GB LPDDR2 SDRAM.
- Ubuntu Server 22.04.1.
- PyTorch 1.8.2.

Table 2 presents the values of β recorded.

As shown in Table 2, there is a large difference in the minibatch runtimes between the tasks. This is due to the relative computational costs of the models used: the Sent140 task uses a simple linear model, whereas the Shakespeare GRU model requires a far larger number of matrix multiplications for a single forward-backward pass.

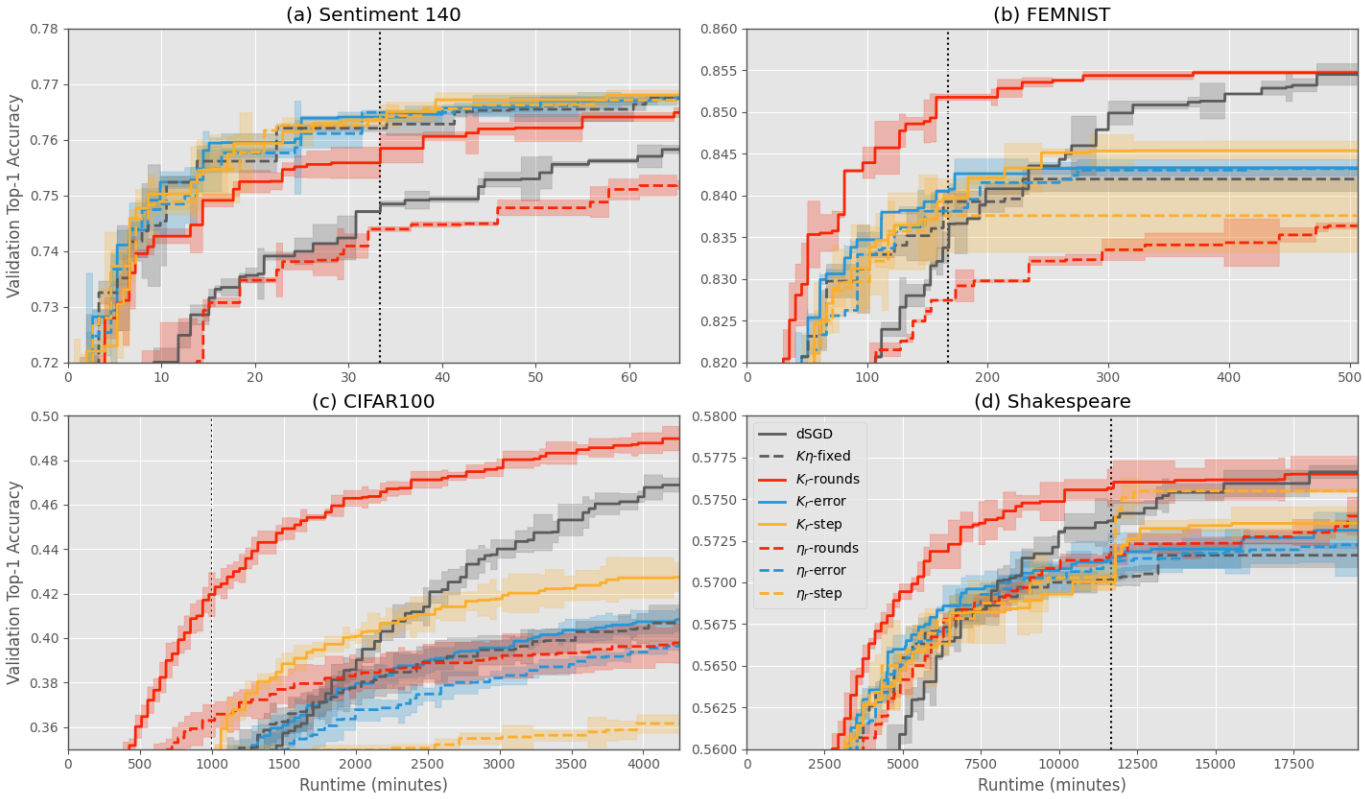

Fig. 2: Cumulative highest validation top-1 accuracy over time of FedAvg using different schedules for K_r and η_r. Curves show mean and shaded regions show 95% confidence intervals of the mean over 5 random trials.

TABLE 2: Mean and standard deviation of runtimes for a mini-batch of SGD for each learning task using a Raspberry Pi 3B+.

Task	Mean (s)	Std (s)
Sent140	5.2×10^{-3}	2.1×10^{-4}
FEMNIST	0.017	5.1×10^{-4}
CIFAR100	0.31	1.7×10^{-2}
Shakespeare	1.5	8.5×10^{-2}

4.3 K_r and η_r Decay Schedules

For each learning task, we ran FedAvg for 10k communication rounds using fixed $K_r = K_0$ and $\eta_r = \eta_0$ (henceforth '$K\eta$-fixed'). The number of rounds reflects typical real-world deployments (which are on the order of thousands of rounds) [25]. We selected K_0 and η_0 via grid-search such that the validation error for each task could plateau within the 10k rounds, and present the values in Table 1. We also ran dSGD (FedAvg with $K_r = 1$) to show the runtime benefit of using $K > 1$ local steps.

We then ran FedAvg using the three schedules for K_r and the three schedules for η_r as discussed in Section 3.4 and 3.5. Table 3 shows the different decay schedules tested and the name we denote each one by in Section 4.4. We also tested jointly decaying K_r *and* η_r during training. However decaying either K_r or η_r decreases the amount of progress that is made during each training round as the global model changes less. We found empirically that decaying both lead to training progress slowing too rapidly, so have not included the results in Section 4.4.

TABLE 3: Values of K_r and η_r for given communication round r as tested in the experimental evaluation.

Schedule	K_r	η_r
dSGD	1	η_0
$K\eta$-fixed	K_0	η_0
K_r-rounds (10)	$\lceil \sqrt[3]{1/r}\ K_0 \rceil$	η_0
K_r-error (13)	$\lceil \sqrt[3]{F_r/F_0}\ K_0 \rceil$	η_0
K_r-step	$K_0/10$ if converged	η_0
η_r-rounds (12)	K_0	$\sqrt[2]{1/r}\ \eta_0$
η_r-error (14)	K_0	$\sqrt[2]{F_r/F_0}\ \eta_0$
η_r-step	K_0	$\eta_0/10$ if converged

4.4 Results

Figure 1 shows the minimum cumulative training error achieved by FedAvg for the different K_r and η_r schedules (as shown in Table 3). Confidence intervals for Figure 1 were omitted for clarity due to the larger number of curves. For all tasks other than Shakespeare, FedAvg with $K\eta$-fixed (solid grey curve) increases the convergence rate compared to dSGD (dashed grey curve). For Shakespeare (Figure 1 (d)), $K\eta$-fixed improved the initial convergence rate but was overtaken by dSGD at approximately 2500 minutes. This is likely because of the very high computation time for Shakespeare (see Table 2) relative to the other datasets (due to the very high computational cost of the GRU model).

For Sentiment 140 (Figure 1 (a)) and Shakespeare (Figure 1

TABLE 4: Total SGD steps performed during training for each K-decay schedule relative to $K\eta$-fixed for different learning tasks.

Task	Schedule	Relative SGD Steps
Sentiment 140	K_r-rounds	0.21
	K_r-error	0.99
	K_r-step	0.68
FEMNIST	K_r-rounds	0.11
	K_r-error	0.80
	K_r-step	0.44
CIFAR100	K_r-rounds	0.090
	K_r-error	0.57
	K_r-step	0.40
Shakespeare	K_r-rounds	0.74
	K_r-error	0.99
	K_r-step	0.96

(d)), decaying either K_r or η_r during training lead to smaller improvements in the training error that was achieved. However, for FEMNIST and CIFAR100, the K_r-rounds scheme lead to lower training error compared to $K\eta$-fixed. For CIFAR100, an improvement was also seen with η_r-rounds. Both FEMNIST and CIFAR100 are image classification tasks, so it be may the case that decaying K_r or η_r during training is beneficial for computer vision tasks, which could be investigated further in future works.

Figure 2 shows the impact on validation accuracy for the tested decay schedules. The $K\eta$-fixed schedule shows faster initial convergence for all tasks, but it is overtaken by dSGD in the later stages of training. For FEMNIST, CIFAR100 and Shakespeare, the aggressive K_r-rounds and K_r-step schemes improved the convergence rate compared to dSGD, with very significant improvement for CIFAR100. A marked increase in convergence rate can be seen in Figure 1 (c) at 1000 minutes when K_r-step is decayed.

In all tasks, all K-decay schemes were able to match or improve the validation accuracy that $K\eta$-fixed achieved whilst performing (often substantially) fewer total steps of SGD within a given runtime. Table 4 shows the total SGD steps performed by the K-decay schemes relative to the total steps performed by $K\eta$-fixed over the 10k communication rounds (all the η-decay schemes perform the same amount of computation as $K\eta$-fixed). The fact that K-decay schemes can outperform $K\eta$fixed with lower total computation indicates that much of the extra computation performed by FedAvg is wasted when considering validation performance. CIFAR100 using K_r-rounds for example achieved over 18% higher validation accuracy compared to $K\eta$-fixed whilst performing less than 10% of the total steps of SGD. Similarly, K_r-step achieved the same validation accuracy as $K\eta$-fixed whilst performing only 68% of the total SGD steps.

5 Conclusion

The popular Federated Averaging (FedAvg) algorithm is used within the Federated Learning (FL) paradigm to improve the convergence rate of an FL model by performing several steps of SGD (K) locally during each training round. In this paper, we analysed FedAvg to examine the runtime benefit of decreasing (K) during training. We set up a runtime model of FedAvg and used this to determine the optimal value of K (and learning rate η) at any point during training under different assumptions, leading to three practical schedules for decaying K as training progresses. Simulated experiments using realistic values for communication-time and computation-time on 4 benchmark FL datasets from 3 learning domains showed that decaying K during training can lead to improved training error and validation accuracy within a given timeframe, in some cases whilst performing over $10\times$ less computation compared to fixed K.

6 Acknowledgements

This work was supported in part by EPSRC New Horizons Grant No. EP/X019160/1, UKRI Grant No. EP/X038866/1, EPSRC DTP studentship, and Horizon EU Grant No. 101086159.

References

[1] P. Kairouz, H. B. McMahan *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1-2, pp. 1–210, 2021.

[2] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv e-prints arXiv:1811.03604*, 2018.

[3] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6341–6345, 2019.

[4] X. Qu, S. Wang, Q. Hu, and X. Cheng, "Proof of federated learning: A novel energy-recycling consensus algorithm," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 8, pp. 2074–2085, 2021.

[5] M. Sheller, B. Edwards, G. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. Colen, and S. Bakas, "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Scientific Reports*, vol. 10, pp. 1–12, 12 2020.

[6] W. Yang, Y. Zhang, K. Ye, L. Li, and C.-Z. Xu, "Ffd: A federated learning based method for credit card fraud detection," in *Big Data – BigData 2019*. Springer International Publishing, 2019, pp. 18–32.

[7] B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *International Conference on Artifical Intelligence and Statistics (AISTATS)*, pp. 1273–1282, 2017.

[8] X. Qiu, T. Parcollet, D. J. Beutel, T. Topal, A. Mathur, and N. D. Lane, "Can federated learning save the planet?" in *NeurIPS - Tackling Climate Change with Machine Learning*, 2020.

[9] C. Wang, Y. Yang, and P. Zhou, "Towards efficient scheduling of federated mobile devices under computational and statistical heterogeneity," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 394–410, 2021.

[10] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986–5994, 2020.

[11] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning (ICML)*, vol. 119, pp. 5132–5143, 2020.

[12] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations (ICLR)*, 2020.

[13] Z. Charles and J. Konečný, "Convergence and accuracy trade-offs in federated learning and meta-learning," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 130, 2021, pp. 2575–2583.

[14] G. Malinovskiy, D. Kovalev, E. Gasanov, L. Condat, and P. Richtarik, "From local SGD to local fixed-point methods for federated learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 119, 2020, pp. 6692–6701.

[15] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-IID federated learning," in *International Conference on Learning Representations (ICLR)*, 2021.

[16] C. T. Dinh, N. H. Tran, M. N. H. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 398–409, 2021.

[17] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems (MLSys)*, vol. 2, pp. 429–450, 2020.
[18] S. P. Karimireddy, M. Jaggi, S. Kale, M. Mohri, S. Reddi, S. U. Stich, and A. T. Suresh, "Breaking the centralized barrier for cross-device federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 28 663–28 676.
[19] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *International Conference on Learning Representations (ICLR)*, 2021.
[20] J. Mills, J. Hu, and G. Min, "Client-side optimization strategies for communication-efficient federated learning," *IEEE Communications Magazine*, vol. 60, no. 7, pp. 60–66, 2022.
[21] B. Woodworth, K. K. Patel, S. Stich, Z. Dai, B. Bullins, B. Mcmahan, O. Shamir, and N. Srebro, "Is local SGD better than minibatch SGD?" in *International Conference on Machine Learning (ICML)*, vol. 119, pp. 10 334–10 343, 2020.
[22] J. Wang and G. Joshi, "Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms," *Journal of Machine Learning Research (JMLR)*, vol. 22, no. 213, pp. 1–50, 2021.
[23] T. Lin, S. U. Stich, K. K. Patel, and M. Jaggi, "Don't use large mini-batches, use local sgd," in *International Conference on Learning Representations (ICLR)*, 2020.
[24] T. Lin, L. Kong, S. Stich, and M. Jaggi, "Extrapolation for large-batch training in deep learning," in *International Conference on Machine Learning (ICML)*, vol. 119, 2020, pp. 6094–6104.
[25] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný *et al.*, "Towards federated learning at scale: System design," in *Proc. Conference on Machine Learning and Systems (SysML)*, 2019.
[26] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, J. Jin, H. Yu, and K. S. Ng, "Towards fair and privacy-preserving federated deep models," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 11, pp. 2524–2541, 2020.
[27] J. Wang and G. Joshi, "Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD," in *Proceedings of Machine Learning and Systems (MLSys)*, vol. 1, pp. 212–229, 2019.
[28] S. Caldas, P. Wu, T. Li, J. Konecný, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," in *NeurIPS Workshop on Federated Learnign for Data Privacy and Confidentiality*, 2019.
[29] "United kingdom mobile network experience report," Open Signal, Tech. Rep., 9 2021, accessed 01/08/2022. [Online]. Available: https://www.opensignal.com/reports/2021/09/uk/mobile-network-experience

Jed Mills is a Computer Science Ph.D. student in the Department of Computer Science at the University of Exeter, UK. He received a B.Sc. in Natural Science from the University of Exeter in 2018. His research interests include machine learning, federated learning and mobile edge computing.

Jia Hu is a Senior Lecturer in Computer Science at the University of Exeter. He received his Ph.D. degree in Computer Science from the University of Bradford, UK, in 2010, and M.Eng. and B.Eng. degrees in Electronic Engineering from Huazhong University of Science and Technology, China, in 2006 and 2004, respectively. His research interests include edge-cloud computing, resource optimization, applied machine learning, and network security.

Geyong Min is a Professor of High Performance Computing and Networking in the Department of Computer Science at the University of Exeter, United Kingdom. He received his Ph.D. degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. His research interests include Computer Networks, Wireless Communications, Parallel and Distributed Computing, Ubiquitous Computing, Multimedia Systems, Modelling and Performance Engineering.

APPENDIX A
PROOF OF THEOREMS

A.1 Key Lemmas

Previously, Li *et al.* [12] analysed the per-iteration convergence of FedAvg for μ-strongly convex functions when using a decreasing stepsize. Their result was the first to prove convergence for non-IID clients with partial participation. We make assumptions that are at least as strong as Li *et al.*, so can use their intermediary result bounding the distance to the global minimiser when using partial client participation:

Lemma 1: *Given Assumptions 1 - 3, the expected distance between average client model $\bar{\boldsymbol{x}}_t$ and the global minimiser $\boldsymbol{x}^*$ is upper-bounded by:*

$$\mathbb{E}\left[\|\bar{\boldsymbol{x}}_{t+1} - \boldsymbol{x}^*\|^2\right] \leq (1-\eta_t\mu)\mathbb{E}\left[\|\bar{\boldsymbol{x}}_t - \boldsymbol{x}^2\|^2\right] + \eta_t^2\Big(\sum_{c=1}^{C} p_c^2\sigma_c^2 + 6L\Gamma + 8(K_t-1)^2G^2 + \frac{C-N}{N-1}\frac{4}{N}K_t^2G^2\Big). \quad (16)$$

Proof: See Appendix B.3 of [12].

Lemma 2: *Given Assumptions 1 - 3, the sum of expected gradient norms over T iterations of the average client model $\bar{\boldsymbol{x}}_t$ is upper-bounded by:*

$$\sum_{t=1}^{T}\eta_t\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_t)\|^2\right] \leq 2\kappa(\kappa F(\bar{\boldsymbol{x}}_0) - F^*) + \kappa L\Big(\sum_{c=1}^{C} p_c^2\sigma_c^2 + 6L\Gamma + 8(K_t-1)^2G^2 + \frac{4}{N}K_t^2G^2\Big)\sum_{t=1}^{T}\eta_t^2. \quad (17)$$

Proof : Rearranging Lemma 1 and then defining for notational convenience

$$D = \left(\sum_{c=1}^{C} p_c^2\sigma_c^2 + 6L\Gamma + 8(K_t-2)^2G^2 + \frac{C-N}{N-1}\frac{4}{N}K_t^2G^2\right),$$

the recursive definition can be written as:

$$\eta_t\mu\mathbb{E}\left[\|\bar{\boldsymbol{x}}_t - \boldsymbol{x}^*\|^2\right] \leq \mathbb{E}\left[\|\bar{\boldsymbol{x}}_t - \boldsymbol{x}^*\|^2\right] - \mathbb{E}\left[\|\bar{\boldsymbol{x}}_{t+1} - \boldsymbol{x}^*\|^2\right] + \eta_t^2 D. \quad (18)$$

Using Assumption 1 (L-smoothness), we have:

$$\frac{\eta_t\mu}{L^2}\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_t)\|^2\right] \leq \mathbb{E}\left[\|\bar{\boldsymbol{x}}_t - \boldsymbol{x}^*\|^2\right] - \mathbb{E}\left[\|\bar{\boldsymbol{x}}_{t+1} - \boldsymbol{x}^*\|^2\right] + \eta_t^2 D. \quad (19)$$

Summing up the T iterations and telescoping the distance terms gives:

$$\frac{\mu}{L^2}\sum_{t=1}^{T}\eta_t^2\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_t)\|^2\right] \leq \mathbb{E}\left[\|\bar{\boldsymbol{x}}_0 - \boldsymbol{x}^*\|^2\right] - \mathbb{E}\left[\|\bar{\boldsymbol{x}}_T - \boldsymbol{x}^*\|^2\right] + D\sum_{t=1}^{T}\eta_t^2. \quad (20)$$

Using Assumption 1 (L-smoothness) and Assumption 2 (μ-strong convexity) to bound the distance terms now gives:

$$\frac{\mu}{L^2}\sum_{t=1}^{T}\eta_t^2\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_t)\|^2\right] \leq \frac{2}{\mu}\left[F(\bar{\boldsymbol{x}}_0) - F^*\right] - \frac{2}{L}\left[F(\bar{\boldsymbol{x}}_T) - F^*\right] + D\sum_{t=1}^{T}\eta_t^2, \quad (21)$$

which can be simplified by noting that $\mu \leq L$, so that:

$$\frac{\mu}{L^2}\sum_{t=1}^{T}\eta_t^2\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_t)\|^2\right] \leq \frac{2}{\mu}F(\bar{\boldsymbol{x}}_0) - \frac{2}{L}F(\bar{\boldsymbol{x}}_T) + D\sum_{t=1}^{T}\eta_t^2. \quad (22)$$

Multiplying both sides of the inequality by L^2/μ, using the lower-bound $F^* \leq F(\bar{\boldsymbol{x}}_T)$, the definition $\kappa = L/\mu$, and the fact that $\frac{C-N}{N-1} \leq 1$ completes the proof.

A.2 Proof of Theorem 1

The bound on gradient norms given in Lemma 2 uses the global index t that denotes the global SGD step that each client evaluates (irrespective of communication rounds). However, FedAvg clients participate in communication rounds. The values of η_t and K_t are therefore fixed for each communication round. To account for this, Lemma 2 can be reindexed using the given communication round r and local step k: $t = I + k$, where where $I = \sum_{i=1}^{r} K_i$. The total number of communication rounds is R, therefore $T = \sum_{r=1}^{R} K_r$. Using this to reindex Lemma 2:

$$\begin{aligned}
&\sum_{r=1}^{R}\eta_r\sum_{k=1}^{K_r}\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_{I+k})\|^2\right] \\
&\quad\leq 2\kappa(\kappa F(\bar{\boldsymbol{x}}_0) - F^*) + \kappa L\Big(\sum_{c=1}^{C} p_c^2\sigma_c^2 + 6L\Gamma + 8(K_r-1)^2G^2 + \frac{4}{N}K_r^2G^2\Big)\sum_{r=1}^{R}\eta_r^2K_r \\
&\quad\leq 2\kappa\left(F(\bar{\boldsymbol{x}}_0) - F^*\right) + \kappa L\left(\sum_{n=1}^{N} p_n^2\sigma_n^2 + 6L\Gamma\right)\sum_{r=1}^{R}\eta_r^2K_r + 16\kappa LG^2\sum_{r=1}^{R}\eta_r^2K_r^3.
\end{aligned}$$

Diving both sides through by $\sum_{r=1}^{} \eta_r K_r$:

$$\begin{aligned}
&\frac{\sum_{r=1}^{R}\eta_r\sum_{k=1}^{K_r}\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_{I+k})\|^2\right]}{\sum_{r=1}^{R}\eta_rK_r} \\
&\quad\leq \frac{2\kappa\left(F(\bar{\boldsymbol{x}}_0) - F^*\right)}{\sum_{r=1}^{R}\eta_rK_r} + \kappa L\left(\sum_{n=1}^{N} p_n^2\sigma_n^2 + 6L\Gamma\right)\frac{\sum_{r=1}^{R}\eta_r^2K_r}{\sum_{r=1}^{R}\eta_rK_r} + 16\kappa LG^2\frac{\sum_{r=1}^{R}\eta_r^2K_r^3}{\sum_{r=1}^{R}\eta_rK_r}.
\end{aligned}$$

Using a fixed $\eta_r = \eta \leq 1/4L$, then the above inequality can be simplified as:

$$\begin{aligned}\frac{\sum_{r=1}^{R}\sum_{k=1}^{K_r}\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_{I+k})\|^2\right]}{\sum_{r=1}^{R}K_r} &\leq \frac{2\kappa\left(F(\bar{\boldsymbol{x}}_0)-F^*\right)}{\eta\sum_{r=1}^{R}K_r} \\ &+ \eta\kappa L\left(\sum_{n=1}^{N}p_n^2\sigma_n^2+6L\Gamma\right) \\ &+ 16\eta\kappa LG^2\frac{\sum_{r=1}^{R}K_r^3}{\sum_{r=1}^{R}K_r}.\end{aligned}$$

Reindexing using the fact that $\sum_{r=1}^{R}K_r = T$ gives:

$$\begin{aligned}\frac{\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_t)\|^2\right]}{T} &\leq \frac{2\kappa\left(F(\bar{\boldsymbol{x}}_0)-F^*\right)}{\eta T} \\ &+ \eta\kappa L\left[\sum_{n=1}^{N}p_n^2\sigma_n^2+6L\Gamma+16G^2\frac{\sum_{r=1}^{R}K_r^3}{\sum_{r=1}^{R}K_r}\right].\end{aligned}$$

Using $\min\{\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_t)\|^2\right]\} \leq \mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_t)\|^2\right]$ then completes the proof.

A.3 Proof of Theorem 2

We start from the bound on gradient norms using a constant K_w (within a communication round) and η (8):

$$\begin{aligned}\min_{t>t_0}\{\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_{t_0})\|^2\right]\} &\leq \frac{2\kappa(\kappa F(\bar{\boldsymbol{x}}_{t_0})-F^*)}{\eta W K_w}\left[\frac{|\boldsymbol{x}|}{D}+\frac{|\boldsymbol{x}|}{U}+\beta K_w\right] \\ &+ \eta\kappa L\left[\sum_{c=1}^{C}p_c^2\sigma_c^2+6L\Gamma+\left(8+\frac{4}{N}\right)G^2K_w^2\right]. \quad (23)\end{aligned}$$

Taking the first derivative with respect to K gives:

$$\begin{aligned}\frac{d\min_{t>t_0}\{\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_{t_0})\|^2\right]\}}{dK_w} &= \frac{-2\kappa(\kappa F(\bar{\boldsymbol{x}}_{t_0})-F^*)}{\eta W K_w^2}\left[\frac{|\boldsymbol{x}|}{D}+\frac{|\boldsymbol{x}|}{U}+\beta K_w\right] \\ &+ 2\eta\kappa L\left(8+\frac{4}{N}\right)G^2K_w. \quad (24)\end{aligned}$$

Taking the second derivative with respect to K_w gives:

$$\begin{aligned}\frac{d^2\min_{t>t_0}\{\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_{t_0})\|^2\right]\}}{dK_w^2} &= \frac{4\kappa(\kappa F(\bar{\boldsymbol{x}}_{t_0})-F^*)}{\eta W K_w^3}\left[\frac{|\boldsymbol{x}|}{D}+\frac{|\boldsymbol{x}|}{U}+\beta K_w\right] \\ &+ 2\eta\kappa L\left(8+\frac{4}{N}\right)G^2. \quad (25)\end{aligned}$$

Considering $(F(\bar{\boldsymbol{x}}_{t_0})-F^*) > 0$ and all the constants in (25) are > 0, then inspection of (25) shows that the second derivative with respect to K_w is greater than 0, and hence (23) is convex. Solving $d\min_{t>t_0}\{\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_{t_0})\|^2\right]\}/dK_w = 0$ gives Theorem 2.

A.4 Proof of Corollary 2.1

As with the proof of Theorem 2, we start with the bound on gradient norms using a constant K and η_w (within a communication round) given in (8):

$$\begin{aligned}\min_{t>t_0}\{\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_{t_0})\|^2\right]\} &\leq \frac{2\kappa(\kappa F(\bar{\boldsymbol{x}}_{t_0})-F^*)}{\eta_w W K}\left[\frac{|\boldsymbol{x}|}{D}+\frac{|\boldsymbol{x}|}{U}+\beta K\right] \\ &+ \eta_w\kappa L\left[\sum_{c=1}^{C}p_c^2\sigma_c^2+6L\Gamma+\left(8+\frac{4}{N}\right)G^2K^2\right]. \quad (26)\end{aligned}$$

Taking the first derivative with respect to η_w gives:

$$\begin{aligned}\frac{d\min_{t>t_0}\{\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_{t_0})\|^2\right]\}}{d\,\eta_w} &= \frac{-2\kappa(\kappa F(\bar{\boldsymbol{x}}_{t_0})-F^*)}{\eta_w^2 W K}\left[\frac{|\boldsymbol{x}|}{D}+\frac{|\boldsymbol{x}|}{U}+\beta K\right] \\ &+ \kappa L\left[\sum_{c=1}^{C}p_c^2\sigma_c^2+6L\Gamma+\left(8+\frac{4}{N}\right)G^2K^2\right]. \quad (27)\end{aligned}$$

Taking the second derivative with respect to η_w gives:

$$\frac{d^2\min_{t>t_0}\{\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_{t_0})\|^2\right]\}}{d\,\eta_w^2} = \frac{4\kappa(\kappa F(\bar{\boldsymbol{x}}_{t_0})-F^*)}{\eta_w^3 W K}\left[\frac{|\boldsymbol{x}|}{D}+\frac{|\boldsymbol{x}|}{U}+\beta K\right]. \quad (28)$$

Noting that $(F(\bar{\boldsymbol{x}}_{t_0})-F^*) > 0$ and all the constants in (28) are > 0, then inspection of (28) shows that the second derivative with respect to η_w is > 0 and hence (27) is convex. Solving $d\min_{t>t_0}\{\mathbb{E}\left[\|\nabla F(\bar{\boldsymbol{x}}_{t_0})\|^2\right]\}/d\eta_w = 0$ yields Corollary 2.1.