Multi-Agent Optimization and Learning
Lecture 2: Learning with a Fusion Center

Stefan Vlaski[†] and Ali H. Sayed[⋆]

[†]Department of Electrical and Electronic Engineering, Imperial College London, UK
[⋆]Adaptive Systems Laboratory, École Polytechnique Fédérale de Lausanne, Switzerland

IEEE ICASSP 2024 Short Course

Objective of this lecture

- Last lecture we saw how risk minimization arises naturally as a formalized learning problem.
- We also saw how models can be learned through stochastic gradient algorithms.
  - Performance was limited by a trade-off between per-iteration complexity, convergence rate and steady-state performance.
- In this lecture we will see how multiple agents can work together via a fusion center to improve on these trade-offs.
- Key-take aways will be:
  - Linear performance gains in centralized (synchronous) and homogeneous structures.
  - Cost of heterogeneity when agents have differing local data distributions.
  - The effect of partial participation and local updates.

Non-cooperative risk minimization

Let us recall the empirical risk minimization problem:

$$w_k^\star = \arg\min_w J_k(w) = \arg\min_w \frac{1}{N}\sum_{n=1}^{N} Q(w; x_{k,n}) \tag{1}$$

Here, w denote model parameters, $x_{k,n}$ denotes the n-th sample available to agent k, and $Q(w; x_{k,n})$ quantifies the fit of model w to the data $x_{k,n}$. The model $w_k^\star$ is then optimal based on the data available to agent k. We can pursue $w_k^\star$ using the gradient-descent algorithm:

$$w_i = w_{i-1} - \mu \nabla J_k(w_{i-1}) = w_{i-1} - \frac{1}{N}\sum_{n=1}^{N} \nabla Q(w_{i-1}; x_{k,n}) \tag{2}$$

or its stochastic variants introduced the last lecture.

Aggregate Optimization Problems

Instead of pursuing locally optimal models, we can instead define a globally optimal model:

$$w^\star = \arg\min_w J(w) = \arg\min_w \frac{1}{KN} \sum_{k=1}^{K} \sum_{n=1}^{N} Q(w; x_{k,n}) \tag{3}$$

In defining $J(w)$, we are now averaging the loss $Q(w; x_{k,n})$ over both the agent index k and the sample index n, hence aggregating all data across the network. For this reason we refer to $w^\star$ as the globally optimal model. Comparing the local and global objectives (1) and (3), we observe the useful relationship:

$$\boxed{J(w) = \frac{1}{K} \sum_{k=1}^{K} J_k(w)} \tag{4}$$

We refer to problem (3), which is equivalent to (4), as the *aggregate optimization problem*, since it aggregates data from all agents. It is also frequently referred to in the literature as the *consensus optimization problem*, since we are looking for a single model w that fits data across the entire collection of agents optimally.

Centralized Gradient Descent

In the absence of constraints on the exchange of information, we can apply gradient-descent directly to the consensus problem (4) and develop the recursion:

$$\begin{aligned} w_i &= w_{i-1} - \mu \nabla J(w_{i-1}) \\ &= w_{i-1} - \frac{\mu}{K} \sum_{k=1}^{K} \nabla J_k(w_{i-1}) \\ &= w_{i-1} - \frac{\mu}{KN} \sum_{k=1}^{K} \sum_{n=1}^{N} \nabla Q(w_{i-1}; x_{k,n}) \end{aligned} \tag{5}$$

Recursion (5) provides an algorithm for solving the consensus optimization problem, but requires central aggregation of the raw data $x_{k,n}$ in order to compute the aggregate gradient $\frac{1}{KN} \sum_{k=1}^{K} \sum_{n=1}^{N} \nabla Q(w_{i-1}; x_{k,n})$.

Aggregate Objectives for Expected Risk Minimization

As we saw last lecture, we may also be interested in local objectives which take the form of an expected risk:

$$J_k(w) = \mathbb{E}_{\boldsymbol{x}_k} Q(w; \boldsymbol{x}_k) \tag{6}$$

Recall can be viewed as a generalization of the empirical risk minimization problem (1). Indeed, if we define:

$$\boldsymbol{x}_k = \begin{cases} x_{k,1}, & \text{with probability } \frac{1}{N}, \\ x_{k,2}, & \text{with probability } \frac{1}{N}, \\ \vdots & \\ x_{k,N}, & \text{with probability } \frac{1}{N}. \end{cases} \tag{7}$$

we can verify that (6) reduces to (1).

Aggregate Objectives for Expected Risk Minimization

In the case of empirical risk minimization problems, the consensus problem (4) carries a clear interpretation as the aggregate empirical risk the the losses are averaged globally across all data available at all agents. Analogously to (4), we can define a consensus problem for expected local risks (6):

$$J(w) = \frac{1}{K}\sum_{k=1}^{K} J_k(w) = \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x}_k} Q(w; \boldsymbol{x}_k) \tag{8}$$

To develop an interpretation for (8), we introduce a random variable $\boldsymbol{x}$ as a mixture of the local data $\boldsymbol{x}_k$ as:

$$\boldsymbol{x} = \begin{cases} \boldsymbol{x}_1, & \text{with probability } \frac{1}{K}, \\ \boldsymbol{x}_2, & \text{with probability } \frac{1}{K}, \\ \vdots & \\ \boldsymbol{x}_K, & \text{with probability } \frac{1}{K}. \end{cases} \tag{9}$$

We can then verify that $J(w) = \mathbb{E}_{\boldsymbol{x}} Q(w; \boldsymbol{x})$.

Centralized Gradient Descent: Parameter Exchange

Applying gradient-descent directly to the consensus problem (4), we obtain the recursion:

$$w_i = w_{i-1} - \frac{\mu}{K} \sum_{k=1}^{K} \nabla J_k(w_{i-1}) \tag{10}$$

Suppose a fusion center sends the current global estimate w_{i-1} to all agents. Then, each agent can perform a local update by descending along its own local risk function based on its private data:

$$\psi_{k,i} = w_{i-1} - \mu \nabla J_k(w_{i-1}) \tag{11}$$

The locally updated models are sent back to the fusion center, where they are aggregated according to:

$$w_i = \frac{1}{K} \sum_{k=1}^{K} \psi_{k,i} \tag{12}$$

Combining (11) and (12), we can verify that $w_i = w_{i-1} - \frac{\mu}{K} \sum_{k=1}^{K} \nabla J_k(w_{i-1})$.

Centralized Gradient Descent: Gradient Exchange

As an alternative to the distributed implementation (11)–(12), we can implement (10) in a distributed manner by exchanging gradients instead of models. In this setting, at iteration i, the parameter server sends the prior model w_{i-1} to all agents. Each agent computes the local gradient, evaluated at the model w_{i-1}:

$$g_{k,i} = \nabla J_k(w_{i-1}) \tag{13}$$

Each agent then sends back the local gradient to the parameter server, where the update is computed via:

$$w_i = w_{i-1} - \frac{\mu}{K} \sum_{k=1}^{K} g_{k,i} \tag{14}$$

Example: Fusing models for logistic regression

We consider logistic regression problem across K agents with local risk functions given by:

$$J_k(w) = \frac{\rho}{2}\|w\|^2 \ + \ \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-\gamma_{k,n} h_{k,n}^{\mathsf{T}} w}\right) \tag{15}$$

We may then pursue a minimizer to the aggregate risk

$$J(w) = \frac{\rho}{2}\|w\|^2 \ + \ \frac{1}{KN}\sum_{k=1}^{K}\sum_{n=1}^{N} \ln\left(1 + e^{-\gamma_{k,n} h_{k,n}^{\mathsf{T}} w}\right) \tag{16}$$

by sharing local iterates as follows:

$$\psi_{k,i} = (1-\mu\rho)w_{i-1} + \mu\left(\frac{1}{N}\sum_{n=1}^{N}\frac{\gamma_{k,n} h_{k,n}}{1 + e^{\gamma_{k,n} h_{k,n}^{\mathsf{T}} w_{i-1}}}\right) \tag{17}$$

$$w_i = \frac{1}{K}\sum_{k=1}^{K}\psi_{k,i} \tag{18}$$

Convergence of the Centralized Gradient Descent Algorithm

Noting that both distributed implementations are equivalent reformulations of the classical gradient descent recursion on $J(w)$, it follows that we can directly apply the gradient descent convergence guarantee to conclude that the iterates w_i will converge to:

$$w^\star \triangleq \arg\min_{w} \frac{1}{K}\sum_{k=1}^{K} J_k(w) \tag{19}$$

linearly, i.e.:

$$\|w^\star - w_i\|^2 \ \leq \ \lambda \|w^\star - w_{i-1}\|^2 \tag{20}$$

where $\lambda \triangleq 1 - 2\mu\nu + \mu^2\delta^2$ and ν and δ correspond to the strong convexity and smoothness constants of the aggregate cost $J(w)$.

Centralized Stochastic Gradient Algorithms

We will continue to study aggregate optimization problems in a multi-agent systems, but now consider the setting where agents no longer have access to exact gradients, and instead employ local gradient approximations as introduced in the last lecture. We thus consider consensus optimization problems of the form:

$$w^o \triangleq \arg\min_{w} \frac{1}{K}\sum_{k=1}^{K} J_k(w) \tag{21}$$

where the local risks take the form

$$J_k(w) = \mathbb{E}_{\boldsymbol{x}_k} Q(w; \boldsymbol{x}_k) \tag{22}$$

We begin by applying the stochastic gradient algorithm to (21). Given local gradient approximations $\widehat{\nabla J}_k(w)$, we may construct an approximation for the global gradient $\nabla J(w)$ by using

$$\widehat{\nabla J}(w) = \frac{1}{K}\sum_{k=1}^{K} \widehat{\nabla J}_k(w) \tag{23}$$

Gradient Noise Bounds for the Induced Gradient Approximation

Suppose each local approximation $\widehat{\nabla J}_k(w)$ satisfies the zero-mean and noise variance conditions introduced in the last lecture. For simplicity, we restrict ourselves in this section to gradient approximations with $\alpha^2 = \gamma^2 = 0$ resulting in:

$$\mathbb{E}\left\{\widehat{\nabla J}_k(\boldsymbol{w}_{i-1}) | \, \boldsymbol{w}_{i-1}\right\} = \nabla J_k(\boldsymbol{w}_{i-1}) \tag{24}$$

$$\mathbb{E}\left\{\left\|\widehat{\nabla J}_k(\boldsymbol{w}_{i-1}) - \nabla J_k(\boldsymbol{w}_{i-1})\right\|^2 | \, \boldsymbol{w}_{i-1}\right\} \leq \beta_k^2 \|w_k^o - \boldsymbol{w}_{i-1}\|^2 + \sigma_k^2 \tag{25}$$

We index the noise constants β_k^2, σ_k^2 by k to emphasize the fact that different agents may have access to gradient approximations of varying quality. Note that the relative component $\beta_k^2 \|w_k^o - \boldsymbol{w}_{i-1}\|^2$ is measured relative to w_k^o, which denotes the local minimizer:

$$w_k^o \triangleq \arg\min_w J_k(w) \tag{26}$$

Gradient Noise Bounds for the Induced Gradient Approximations

We can verify that (23) is unbiased, i.e.:

$$\mathbb{E}\left\{\widehat{\nabla J}(\boldsymbol{w}_{i-1})|\,\boldsymbol{w}_{i-1}\right\} = \frac{1}{K}\sum_{k=1}^{K}\nabla J_k(\boldsymbol{w}_{i-1}) \tag{27}$$

For the variance, assuming the data is independent between agents, we have:

$$\mathbb{E}\left\{\left\|\widehat{\nabla J}(\boldsymbol{w}_{i-1}) - \nabla J(\boldsymbol{w}_{i-1})\right\|^2 |\,\boldsymbol{w}_{i-1}\right\} \leq \frac{1}{K^2}\sum_{k=1}^{K}\left(\beta_k^2\|w_k^o - \boldsymbol{w}_{i-1}\|^2 + \sigma_k^2\right) \tag{28}$$

We can expand:

$$\mathbb{E}\left\{\left\|\widehat{\nabla J}(\boldsymbol{w}_{i-1}) - \nabla J(\boldsymbol{w}_{i-1})\right\|^2 |\,\boldsymbol{w}_{i-1}\right\} \leq \beta^2\|w^o - \boldsymbol{w}_{i-1}\|^2 + \sigma^2 \tag{29}$$

where:

$$\beta^2 = \frac{1}{K^2}\left(\sum_{k=1}^{K}2\beta_k^2\right), \quad \sigma^2 = \frac{1}{K^2}\sum_{k=1}^{K}\left(2\beta_k^2\|w_k^o - w^o\|^2 + \sigma_k^2\right) \tag{30}$$

Centralized Stochastic Gradient: Parameter Exchange

Each agent receives the current model $\boldsymbol{w}_{i-1}$ from the fusion center and performs a local update using its local gradient approximation:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{i-1} - \mu \widehat{\nabla J}_k(\boldsymbol{w}_{i-1}) \tag{31}$$

The locally updated models are sent to the fusion center, where they are aggregated according to:

$$\boldsymbol{w}_i = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\psi}_{k,i} \tag{32}$$

It can be verified that (31)–(32) is equivalent to stochastic gradient descent on the aggregate objective, while requiring only the exchange of intermediate models $\boldsymbol{\psi}_{k,i}$.

Centralized Stochastic Gradient: Gradient Exchange

We can also implement the stochastic gradient algorithm in a distributed manner by instead exchanging gradient approximations. Again, at iteration i, the parameter server sends the prior model $\boldsymbol{w}_{i-1}$ to all agents. Each agent computes the local gradient, evaluated at the model $\boldsymbol{w}_{i-1}$:

$$\boldsymbol{g}_{k,i} = \widehat{\nabla J}_k(\boldsymbol{w}_{i-1}) \tag{33}$$

Each agent then sends back the local gradient to the parameter server, where the update is computed via:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \frac{\mu}{K} \sum_{k=1}^{K} \boldsymbol{g}_{k,i} \tag{34}$$

It is again straightforward to verify that the resulting recursion is equivalent stochastic gradient on the aggregate loss. We illustrate a schematic of the centralized stochastic gradient algorithm with model exchanges on the next slide.

Visualization of the Centralized Stochastic Gradient Algorithm

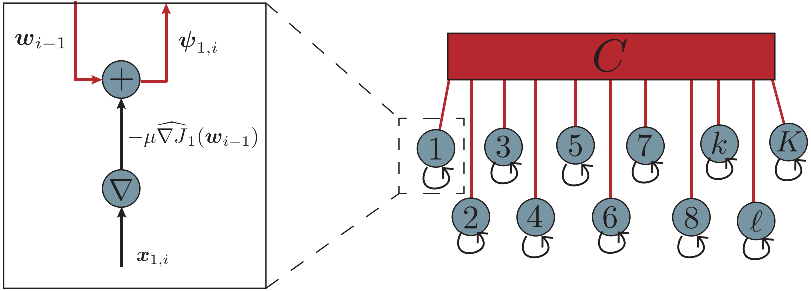


Figure: A schematic representation of the centralized stochastic gradient algorithm (31)–(32) with model exchanges.

Example: Mean-square-error

Suppose each agent observes data arising from a linear regression model of the form:

$$\boldsymbol{\gamma}_k = \boldsymbol{h}_k^{\mathsf{T}} w^o + \boldsymbol{v}_k \tag{35}$$

Note that while the parameter w^o in this example is common to all agents, the random variables $\boldsymbol{h}_k$, $\boldsymbol{v}_k$, and hence $\boldsymbol{\gamma}_k$ may follow distinct distributions. We may then formulate the local risks:

$$J_k(w) = \frac{1}{2}\mathbb{E}\|\boldsymbol{\gamma}_k - \boldsymbol{h}_k^{\mathsf{T}} w\|^2 \tag{36}$$

with elementary gradient approximation given by

$$\widehat{\nabla J}_k(w) = \nabla Q(w; \boldsymbol{h}_{k,i}, \boldsymbol{\gamma}_{k,i}) = -\boldsymbol{h}_{k,i}\left(\boldsymbol{\gamma}_{k,i} - \boldsymbol{h}_{k,i}^{\mathsf{T}} w\right) \tag{37}$$

Example: Mean-Square Error

This leads to a centralized implementation of the stochastic gradient recursion as follows:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{i-1} + \mu \boldsymbol{h}_{k,i} \left(\boldsymbol{\gamma}_{k,i} - \boldsymbol{h}_{k,i}^{\mathsf{T}} \boldsymbol{w}_{i-1} \right) \tag{38}$$

$$\boldsymbol{w}_i = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\psi}_{k,i} \tag{39}$$

Performance of the Centralized Stochastic Gradient Algorithm

Since the stochastic gradient algorithms presented here, whether implemented via parameter or gradient exchange, yield identical iterates, and they correspond to employing the gradient approximation (23), we may infer the resulting performance from the convergence guarantee of stochastic gradient algorithms in the previous lecture. Using (30), we find:

$$\lim_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_i\|^2 \le \frac{\mu}{\nu K^2} \sum_{k=1}^{K} \left(2\beta_k^2 \|w_k^o - w^o\|^2 + \sigma_k^2 \right) \tag{40}$$

Linear Gain for Homogeneous Agents

A useful simplification occurs if we consider the setting of perfectly homogeneous agents. In a homogeneous setting, the data $\boldsymbol{x}_k$ observed by each agent is identically distributed, and all gradient approximations take the same form. It follows that:

$$J_k(w) = \mathbb{E}_{\boldsymbol{x}_k} Q(w; \boldsymbol{x}_k) = \mathbb{E}_{\boldsymbol{x}_\ell} Q(w; \boldsymbol{w}_\ell) \tag{41}$$

for all k, ℓ, and hence $J_k(w) = J(w)$ for all k. Similarly, we have $w_k^o = w^o$ and $\sigma_k^2 = \sigma_1^2$. Then expression (40) simplifies to

$$\lim_{i \to \infty} \mathbb{E} \| w^o - \boldsymbol{w}_i \|^2 \leq \frac{\mu \sigma_1^2}{\nu K} \tag{42}$$

We conclude that the limiting performance of the centralized architecture improves at a rate of $\frac{1}{K}$, where K is the number of agents. This fact is referred to as *linear gain* in distributed systems and is an important motivator for agents to participate in learning.

Numerical Results

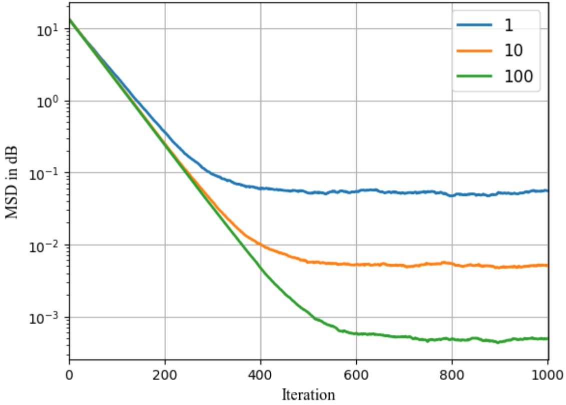


Figure: A demonstration in *linear gain in performance* of the centralized stochastic gradient algorithm. The steady-state error decays by a factor of 10 as the number of participating agents increases by a factor of 10. This is predicted by (42).

Recap and Outlook

- We have seen that we can implement (stochastic) gradient algorithms in a centralized manner by relying only on the exchange of models or gradients in place of raw data.
- For homogeneous agents, this resulted in linear performance gains.
- In practice, it is not desirable to require regular and constant participation of agents.
  - Consider for example wireless sensors or mobile devices with limited power and communication capabilities.
- We now introduce to types of imperfections and study their impact on learning dynamics:
  - Partial participation (asynchronous learning)
  - Local updates (federated learning)

Agent Sampling

Instead of involving all agents at each iteration, as in the case of (31)–(32), we will now study an *asynchronous* variant where at each iteration a single agent is selected, and the parameter is only updated by the selected agent. To make this description more precise, we introduce the random index $\boldsymbol{k}_i$, which denotes the agent picked at time instant i. This index follows a uniform distribution:

$$\boldsymbol{k}_i = \begin{cases} 1, & \text{with probability } \frac{1}{K}, \\ 2, & \text{with probability } \frac{1}{K}, \\ \vdots & \\ K, & \text{with probability } \frac{1}{K}. \end{cases} \tag{43}$$

The fusion center then provides the selected agent $\boldsymbol{k}_i$ with the previous model $\boldsymbol{w}_{i-1}$. Agent $\boldsymbol{k}_i$ updates the model:

$$\boldsymbol{\psi}_{\boldsymbol{k}_i,i} = \boldsymbol{w}_{i-1} - \mu \widehat{\nabla J}_{\boldsymbol{k}_i}(\boldsymbol{w}_{i-1}) \tag{44}$$

Agent Sampling

The sampled agent send it back to the server where the global model is updated to

$$\boldsymbol{w}_i = \boldsymbol{\psi}_{\boldsymbol{k}_i,i} \tag{45}$$

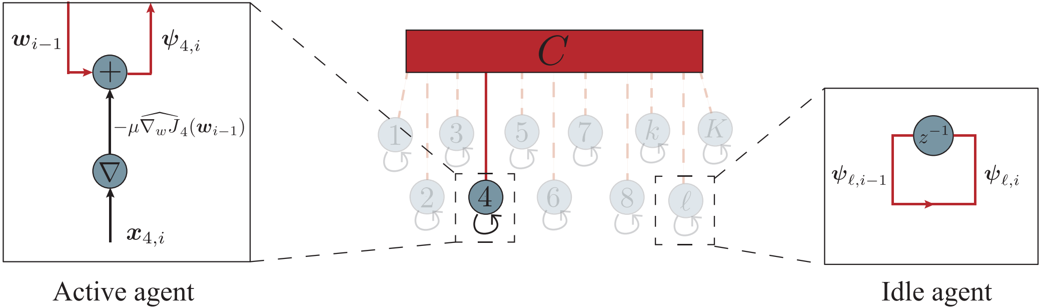
We can write (44)–(45) compactly as:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \widehat{\nabla J}_{\boldsymbol{k}_i}(\boldsymbol{w}_{i-1}) = \boldsymbol{w}_{i-1} - \mu \widehat{\nabla J}(\boldsymbol{w}_{i-1}) \tag{46}$$

where we defined:

$$\widehat{\nabla J}(\boldsymbol{w}_{i-1}) = \widehat{\nabla J}_{\boldsymbol{k}_i}(\boldsymbol{w}_{i-1}) \tag{47}$$

Visualization of an Asynchronous Stochastic Gradient Algorithm



$\boldsymbol{w}_{i-1}$
$\boldsymbol{\psi}_{4,i}$
$+$
$-\mu\widehat{\nabla_w J}_4(\boldsymbol{w}_{i-1})$
∇
$\boldsymbol{x}_{4,i}$
Active agent
C
1
3
5
7
k
K
2
4
6
8
ℓ
z^{-1}
$\boldsymbol{\psi}_{\ell,i-1}$
$\boldsymbol{\psi}_{\ell,i}$
Idle agent

Figure: A schematic representation of the asynchronous stochastic gradient algorithm (44)–(45).

Gradient Noise induced by Agent Sampling

In order to study the performance of this asynchronous algorithm, we need to establish conditions on the gradient noise induced by the approximation (47). We can verify:

$$\mathbb{E}\left\{\widehat{\nabla J}(\boldsymbol{w}_{i-1}) \middle| \, \boldsymbol{w}_{i-1}\right\} = \nabla J(\boldsymbol{w}_{i-1}) \tag{48}$$

$$\mathbb{E}\left\{\left\|\widehat{\nabla J}(\boldsymbol{w}_{i-1}) - \nabla J(\boldsymbol{w}_{i-1})\right\|^2 \middle| \, \boldsymbol{w}_{i-1}\right\} \le \beta^2 \| w^o - \boldsymbol{w}_{i-1} \|^2 + \sigma^2 \tag{49}$$

where defined:

$$\beta^2 = \frac{1}{K}\sum_{k=1}^{K}\left(2\beta_k^2 + 4\delta_k^2 + 2\delta^2\right) \tag{50}$$

$$\sigma^2 = \frac{1}{K}\sum_{k=1}^{K}\left(2\left(\beta_k^2 + 2\delta_k^2\right)\|w_k^o - w^o\|^2 + \sigma_k^2\right) \tag{51}$$

Performance of Asynchronous Stochastic Gradient

From the stochastic gradient theorem, we can then conclude:

$$\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_i\|^2 \leq \frac{\mu}{\nu K}\sum_{k=1}^{K}\left(2\left(\beta_k^2 + 2\delta_k^2\right)\|w_k^o - w^o\|^2 + \sigma_k^2\right) \tag{52}$$

In the homogeneous case where $w_k^o = w^o$, $\sigma_k^2 = \sigma_1^2$, this simplifies to:

$$\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_i\|^2 \leq \frac{\mu\sigma_1^2}{\nu} \tag{53}$$

which is the same performance as a non-cooperative approach. This is to be expected, as only a single agent is participating in the learning protocol at any given iteration.

Cost of Heterogeneity

A second insightful simplification of the performance bound (52) occurs when considering exact gradient approximations $\widehat{\nabla J}_k(\boldsymbol{w}_{i-1}) = \nabla J_k(\boldsymbol{w}_{i-1})$, which guarantees $\beta_k^2 = \sigma_k^2 = 0$. Nevertheless, as long as agents are heterogeneous and $w^o \neq w_k^o$, we find:

$$\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_i\|^2 \leq \frac{\mu}{\nu K}\sum_{k=1}^{K} 4\delta_k^2\|w_k^o - w^o\|^2 \tag{54}$$

where δ_k^2 are the Lipschitz constants of the local true gradients. It follows that even if every agent in the network is employing an exact gradient, the asynchrony of the system introduces a randomness into the recursion of the model, and ultimately results in deterioration of performance at steady-state. This performance deterioration grows with the system heterogeneity, captured in $\|w_k^o - w^o\|^2$, while vanishing for homogeneous systems. We will see this theme appear repeatedly in future lectures of this course when studying federated and fully decentralized structures.

Within- and cross-agent variance

The level of heterogeneity, which is measured by $\frac{1}{K}\sum_{k=1}^{K} 4\delta_k^2\|w_k^o - w^o\|^2$, is sometimes referred to in the literature as *cross*-agent or *inter*-agent variance. This is to distinguish it from the *within*- or *intra*-agent variance in the local gradient approximations $\widehat{\nabla J}_k(\boldsymbol{w}_{i-1})$, which corresponds to the variance of the local gradient noise, and is a function of the local data distributions $\boldsymbol{x}_k$. Strictly speaking, since the quantities w_k^o and w^o are generally deterministic and fixed a priori, the term "variance" a misnomer, and "variation" or "heterogeneity" are more accurate. If we interpret w_k^o as samples from a common distribution with mean w^o, then $\frac{1}{K}\sum_{k=1}^{K} 4\delta_k^2\|w_k^o - w^o\|^2$ can more accurately be interpreted as a weighted, finite-sample approximation of variance of the underlying generating distribution.

From Centralized to Federated Learning

A common key property the centralized algorithms we have studied so far is that in all cases:

$$\mathbb{E}\left\{\boldsymbol{w}_i \,|\, \boldsymbol{w}_{i-1}\right\} = \boldsymbol{w}_{i-1} - \mu \nabla J(\boldsymbol{w}_{i-1}) = \boldsymbol{w}_{i-1} - \frac{\mu}{K} \sum_{k=1}^{K} \nabla J_k(\boldsymbol{w}_{i-1}) \tag{55}$$

We exploited this fact up until now to develop performance guarantees for a number of stochastic centralized algorithms by simply quantifying that variance of the induced gradient noise process, and subsequently appealing to standard convergence guarantees for stochastic gradient descent.

- We will now allow for more general subsampling of L out of K agents.
- Additionally, we allow agents to take multiple local update steps in-between communication exchanges, which is characteristic of federated learning. This causes (55) to no longer hold.

Federated Averaging

The parameter server continues to maintain a global model $\boldsymbol{w}_i$. At every iteration i, only a subset of L agents, collected in $\boldsymbol{\mathcal{L}}_i$ will be selected to participate. We will generate the set $\boldsymbol{\mathcal{L}}_i$ by sampling uniformly with equal probability and *without replacement*, so that:

$$\mathbb{P}\left\{k \in \boldsymbol{\mathcal{L}}_i\right\} = \frac{L}{K} \tag{56}$$

Now, at time i, each selected agent $k \in \boldsymbol{\mathcal{L}}_i$ receives the model $\boldsymbol{w}_{i-1}$ from the parameter server, initializes $\boldsymbol{\phi}_{k,0} = \boldsymbol{w}_{i-1}$ and then performs E_k local updates of the form:

$$\boldsymbol{\phi}_{k,e} = \boldsymbol{\phi}_{k,e-1} - \mu_k \widehat{\nabla J_k}(\boldsymbol{\phi}_{k,e-1}) \tag{57}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,E_k} \tag{58}$$

In (57) we utilize e to index the inner iteration of local gradient updates, which occurs between every outer time step $i-1$ and i.

Federated Averaging

The intermediate models are sent back to the server, where aggregation takes the form:

$$\boldsymbol{w}_i = \frac{1}{L} \sum_{k \in \boldsymbol{\mathcal{L}}_i} \boldsymbol{\psi}_{k,i} \tag{59}$$

Before we proceed with studying the learning dynamics of the federated algorithm (57)–(59), it is important to comment on some important details in the recursions. First, we allow for differing numbers of local updates E_k at different agents. This is important to account for different capabilities across different agents. Some agents may be able to perform more local updates in a given period of time, for example due to increased computational capabilities. Second, we allow for varying local step-size parameters μ_k. This additional degree of freedom will be necessary to control the relative influence of agents with different number of local updates E_k.

High-Level Insight

While we will present a formal guarantee of convergence of the Federated Averaging algorithm (57)–(59) further ahead, we will first reformulate the recursions in an equivalent form that is less amenable to distributed implementation, but provides a high-level intuition behind the learning dynamics of federated averaging. These insights will serve two purposes. First, it will suggest a choice for the local step-sizes μ_k as a function of the number of local updates E_k. Second, it will provide a sketch for the convergence analysis that leads to the theorem we present later.

Iterating (57) and plugging the final result into (57), we can find for the locally adapted models $\boldsymbol{\psi}_{k,i}$:

$$\begin{aligned} \boldsymbol{\psi}_{k,i} &= \boldsymbol{w}_{i-1} - \mu_k \sum_{e=1}^{E_k} \widehat{\nabla J}_k(\boldsymbol{\phi}_{k,e-1}) \\ &= \boldsymbol{w}_{i-1} - \mu_k E_k \cdot \frac{1}{E_k} \sum_{e=1}^{E_k} \widehat{\nabla J}_k(\boldsymbol{\phi}_{k,e-1}) \end{aligned} \tag{60}$$

High-Level Insight

Inspection of this form reveals that (60) resembles a mini-batch update with increased step-size $\mu_k E_k$, where we are averaging E_k mini-batch approximations $\widehat{\nabla J_k}(\boldsymbol{\phi}_{k,e-1})$. One key detail to note, however, is that the gradient approximations $\widehat{\nabla J_k}(\boldsymbol{\phi}_{k,e-1})$ are all evaluated at different iterates $\boldsymbol{\phi}_{k,e-1}$, hence

$$\frac{1}{E_k}\sum_{e=1}^{E_k}\widehat{\nabla J_k}(\boldsymbol{\phi}_{k,e-1}) \neq \frac{1}{E_k}\sum_{e=1}^{E_k}\widehat{\nabla J_k}(\boldsymbol{w}_{i-1}) \tag{61}$$

and we are not performing a true mini-batch gradient update. Instead, we can interpret the federated local update as a perturbed mini-batch update by reformulating:

$$\begin{aligned}\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{i-1} &- \mu_k E_k \cdot \frac{1}{E_k}\sum_{e=1}^{E_k}\widehat{\nabla J_k}(\boldsymbol{w}_{i-1}) \\ &+ \mu_k E_k \cdot \frac{1}{E_k}\sum_{e=1}^{E_k}\left(\widehat{\nabla J_k}(\boldsymbol{w}_{i-1}) - \widehat{\nabla J_k}(\boldsymbol{\phi}_{k,e-1})\right)\end{aligned} \tag{62}$$

Step-Size Normalization for Federated Averaging

To develop appropriate expressions for the local step-sizes μ_k, we first introduce the following quantities for ease of notation:

$$\widehat{\nabla J_k}^{E_k}(\boldsymbol{w}_{i-1}) \triangleq \frac{1}{E_k} \sum_{e=1}^{E_k} \widehat{\nabla J_k}(\boldsymbol{w}_{i-1}) \tag{63}$$

$$\boldsymbol{d}_{k,i} = \frac{1}{E_k} \sum_{e=1}^{E_k} \left(\widehat{\nabla J_k}(\boldsymbol{\phi}_{k,e-1}) - \widehat{\nabla J_k}(\boldsymbol{w}_{i-1}) \right) \tag{64}$$

We can then write (60) more compactly:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{i-1} - \mu_k E_k \widehat{\nabla J_k}^{E_k}(\boldsymbol{w}_{i-1}) - \mu_k E_k \boldsymbol{d}_{k,i} \tag{65}$$

in terms of the mini-batch gradient approximation $\widehat{\nabla J_k}^{E_k}(\boldsymbol{w}_{i-1})$ and the perturbation $\boldsymbol{d}_{k,i}$.

Step-Size Normalization for Federated Averaging

After aggregation following (59), we then have at the parameter server:

$$\begin{aligned}\boldsymbol{w}_i &= \frac{1}{L}\sum_{k\in\boldsymbol{\mathcal{L}}_i}\left(\boldsymbol{w}_{i-1} - \mu_k E_k \widehat{\nabla J_k}^{E_k}(\boldsymbol{w}_{i-1}) - \mu_k E_k \boldsymbol{d}_{k,i}\right)\\ &= \boldsymbol{w}_{i-1} - \frac{1}{L}\sum_{k\in\boldsymbol{\mathcal{L}}_i}\mu_k E_k \widehat{\nabla J_k}^{E_k}(\boldsymbol{w}_{i-1}) - \frac{1}{L}\sum_{k\in\boldsymbol{\mathcal{L}}_i}\mu_k E_k \boldsymbol{d}_{k,i}\\ &\overset{(a)}{=} \boldsymbol{w}_{i-1} - \frac{\mu}{L}\sum_{k\in\boldsymbol{\mathcal{L}}_i}\frac{\mu_k}{\mu} E_k \widehat{\nabla J_k}^{E_k}(\boldsymbol{w}_{i-1}) - \frac{1}{L}\sum_{k\in\boldsymbol{\mathcal{L}}_i}\mu_k E_k \boldsymbol{d}_{k,i} \qquad (66)\end{aligned}$$

where in (a) we introduced an arbitrary common step-size $\mu > 0$, and normalized the local step-sizes inside the sum by the same factor μ, leaving the recursion unchanged.

Step-Size Normalization for Federated Averaging

We can then introduce the global quantities:

$$\widehat{\nabla J}^{\text{fed}}(\boldsymbol{w}_{i-1}) \triangleq \frac{1}{L} \sum_{k \in \boldsymbol{\mathcal{L}}_i} \frac{\mu_k}{\mu} E_k \widehat{\nabla J_k}^{E_k}(\boldsymbol{w}_{i-1}) \tag{67}$$

$$\boldsymbol{d}_i \triangleq \frac{1}{L} \sum_{k \in \boldsymbol{\mathcal{L}}_i} \mu_k E_k \boldsymbol{d}_{k,i} \tag{68}$$

and write more compactly:

$$\begin{aligned} \boldsymbol{w}_i &= \boldsymbol{w}_{i-1} - \mu \widehat{\nabla J}^{\text{fed}}(\boldsymbol{w}_{i-1}) - \mu \boldsymbol{d}_i \\ &= \boldsymbol{w}_{i-1} - \mu \nabla J(\boldsymbol{w}_{i-1}) - \mu\, \boldsymbol{s}_i(\boldsymbol{w}_{i-1}) - \mu \boldsymbol{d}_i \end{aligned} \tag{69}$$

with the gradient noise term $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$ defined as before:

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) = \widehat{\nabla J}^{\text{fed}}(\boldsymbol{w}_{i-1}) - \nabla J(\boldsymbol{w}_{i-1}) \tag{70}$$

Step-Size Normalization for Federated Averaging

One of the requirements we have so far placed on gradient approximations is that they are unbiased, which implies that the gradient noise process $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$ they induce has mean zero. To ensure:

$$\mathbb{E}\left\{\widehat{\nabla J}^{\text{fed}}(\boldsymbol{w}_{i-1})\middle|\,\boldsymbol{w}_{i-1}\right\} = \nabla J(\boldsymbol{w}_{i-1}) \tag{71}$$

we need to set:

$$\boxed{\frac{\mu_k}{\mu} E_k = 1 \Longleftrightarrow \mu_k = \frac{\mu}{E_k}} \tag{72}$$

Mean-square-behavior of federated averaging

Let the individual objectives $J_k(w)$ be ν_k-strongly convex and δ_k-smooth, and the aggregate objective $J(w)$ be ν-strongly convex with δ-Lipschitz gradients. Suppose further that the local gradient approximations $\widehat{\nabla J_k}(\cdot)$ with constants $\alpha_k^2 = \gamma_k^2 = 0$ and $\beta_k^2, \sigma_k^2 \geq 0$. Then, the error $\widetilde{\boldsymbol{w}}_i \triangleq w^o - \boldsymbol{w}_i$ of the iterates generated by the federated averaging algorithm (57)–(59) satisfy:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \left(\sqrt{\lambda} + O(\mu^3)\right) \mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 + 2\mu^2 \sigma_{\text{fed}}^2 + O(\mu^3) \tag{73}$$

where

$$\sqrt{\lambda} = \sqrt{1 - 2\mu\nu + \mu^2\left(\delta^2 + \beta_{\text{fed}}^2\right)} \leq 1 - \mu\nu + \frac{\mu^2}{2}\left(\delta^2 + \beta_{\text{fed}}^2\right) \tag{74}$$

Then, for sufficiently small step-sizes it holds that $\sqrt{\lambda} + O(\mu^3) < 1$ and we can iterate this relation to find:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq (\sqrt{\lambda} + O(\mu^3))^i \|\widetilde{w}_0\|^2 + \frac{4\mu\sigma_{\text{fed}}^2}{\nu} + O(\mu^2) \tag{75}$$

Homogeneous Settings

It is instructive to simplify the performance expression this theorem to illustrate the benefit of employing local updates, and allowing for partial participation. In particular, we know from (75), that the limiting performance of the federated averaging algorithm is given by:

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq O\left(\frac{\mu\sigma_{\text{fed}}^2}{\nu}\right) + O(\mu^2) \tag{76}$$

where we used the $O(\cdot)$ notation to remove multiplying constants and higher-order terms in the step-size. The key quantity in this expression, analogously to the stochastic gradient algorithms seen earlier, is the absolute gradient noise power σ_{fed}^2. We have:

$$\sigma_{\text{fed}}^2 = \frac{1}{KL}\sum_{k=1}^{K}\left(2\frac{\beta_k^2}{E_k}\|w_k^o - w^o\|^2 + \frac{\sigma_k^2}{E_k}\right) + \frac{1}{KL}\frac{K-L}{K-1}\sum_{k=1}^{K} 4\delta_k^2\|w_k^o - w^o\|^2 \tag{77}$$

Homogeneous Settings

When agents are observing data following the same distribution, it follows that $w_k^o = w^o$, and hence:

$$\sigma_{\text{fed}}^2 = \frac{1}{KL} \sum_{k=1}^{K} \frac{\sigma_k^2}{E_k} \tag{78}$$

Let us further assume that the gradient approximations constructed by each agent are identical, so $\sigma_k^2 = \sigma^2$ and $E_k = E$. Then:

$$\sigma_{\text{fed}}^2 = \frac{1}{KL} \sum_{k=1}^{K} \frac{\sigma^2}{E} = \frac{1}{L} \frac{\sigma^2}{E} \tag{79}$$

Hence, the limiting performance of the federated averaging algorithm in this homogeneous case is given by:

$$\limsup_{i \to \infty} \mathbb{E} \|\widetilde{\boldsymbol{w}}_i\|^2 \leq O\left(\frac{\mu \sigma^2}{\nu L E}\right) + O(\mu^2) \tag{80}$$

Cost of Heterogeneity

Similarly to before, we can simplify to the heterogeneous setting with exact gradients by letting $\widehat{\nabla J}_k(\boldsymbol{w}_{i-1}) = \nabla J_k(\boldsymbol{w}_{i-1})$, which guarantees $\beta_k^2 = \sigma_k^2 = 0$. Nevertheless, as long as agents are heterogeneous and $w^o \neq w_k^o$, we find:

$$\sigma_{\text{fed}}^2 = \frac{1}{KL} \frac{K-L}{K-1} \sum_{k=1}^{K} 4\delta_k^2 \|w_k^o - w^o\|^2 \tag{81}$$

This means that again the subsampling of agents, when they are heterogeneous, can lead to deterioration in performance. This can be addressed by adapting variance reduction techniques to the federated setting, see, e.g., SCAFFOLD [Karimireddy et al, 2020].

Lab #1: Learning with a Fusion Center

We will now present a simulation study that allows us to illustrate some of the key-take aways from the first two lectures. Jupyter notebooks to follow along and play with parameters are provided on the course website, although participants are encouraged to develop their own code base. Specifically, we will provide a collection of K agents with data following the linear model:

$$\boldsymbol{\gamma}_k = \boldsymbol{h}_k^{\mathsf{T}} w_k^o + \boldsymbol{v}_k \tag{82}$$

with isotropic regressors $\boldsymbol{h}_k \sim \mathcal{N}(0, \sigma_h^2 I_M) \in \mathbb{R}^M$ and Gaussian noise $\boldsymbol{v}_k \sim \mathcal{N}(0, \sigma_v^2) \in \mathbb{R}$.

Controlling Heterogeneity

To control the heterogeneity of the local models w_k^o, we sample them from the distribution $\mathcal{N}(\mathbb{1}, \sigma_w^2 I_M) \in \mathbb{R}^M$. In this manner, by setting $\sigma_w^2 = 0$, we recover a homogeneous data setting with $w_k^o = w^o = \mathbb{1}$, while $\sigma_w^2 > 0$ results in heterogeneous models with variance defined by σ_w^2. Each local loss is given by:

$$J_k(w) = \frac{1}{2}\mathbb{E}\|\boldsymbol{\gamma}_k - \boldsymbol{h}_k^\mathsf{T} w\|^2 \tag{83}$$

It can then be verified that $w_k^o = \arg\min_w J_k(w)$ and $w^o = \arg\min_w J(w)$, where $J(w) = \frac{1}{K}\sum_{k=1}^K J_k(w)$. Since agents have common regressor covariances R_h, it can be seen from the optimality conditions that $w^o = \frac{1}{K}\sum_{k=1}^K w_k^o$. Each agent constructs local gradient approximations:

$$\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) = -\boldsymbol{h}_{k,i}\left(\boldsymbol{\gamma}_{k,i} - \boldsymbol{h}_{k,i}^\mathsf{T}\boldsymbol{w}_{k,i-1}\right) \tag{84}$$

Fusion-Center Based Algorithms

The federated averaging algorithm in this setting amounts to:

$$\boldsymbol{\phi}_{k,e} = \boldsymbol{\phi}_{k,e-1} + \frac{\mu}{E}\boldsymbol{h}_{k,i}\left(\boldsymbol{\gamma}_{k,i} - \boldsymbol{h}_{k,i}^{\mathsf{T}}\boldsymbol{\phi}_{k,e-1}\right), \qquad e = 1, \ldots, E \tag{85}$$

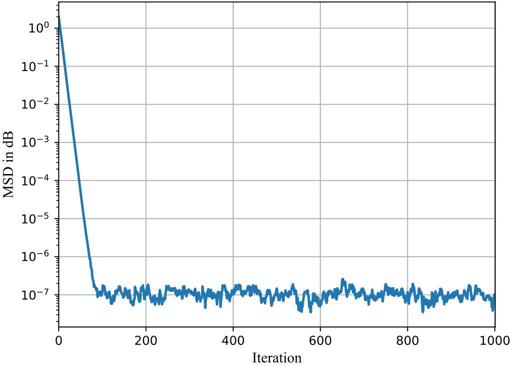
$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,E_k} \tag{86}$$

$$\boldsymbol{w}_i = \frac{1}{L}\sum_{k\in\boldsymbol{\mathcal{L}}_i}\boldsymbol{\psi}_{k,i} \tag{87}$$

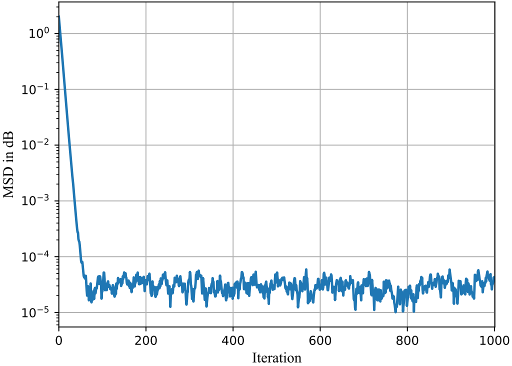
Note that by setting $L = K, E = 1$, we recover the centralized stochastic gradient algorithm with full participation, while $L = 1, E = 1$ recovers the asynchronous variant where one agent is sampled.

Observation #1: Heterogeneity hurts Performance even of Centralized Stochastic Gradient

- Consistent with $\sigma^2_{\text{cent}} = \frac{1}{K^2}\sum_{k=1}^{K}\left(2\beta_k^2\|w_k^o - w^o\|^2 + \sigma_k^2\right)$.



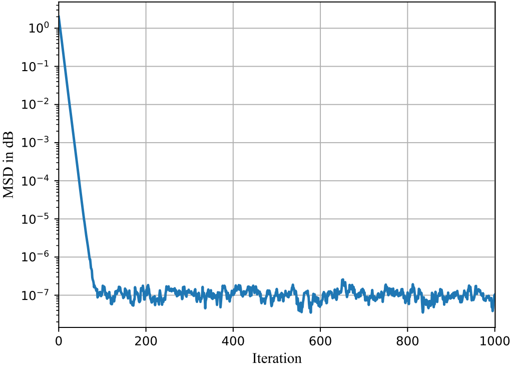
$$K = L = 100,\ E = 1,\ \sigma_w^2 = 0$$



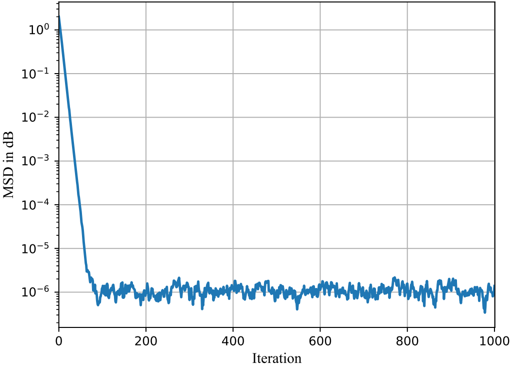
$$K = L = 100,\ E = 1,\ \sigma_w^2 = 0.1$$

Observation #2: For homogeneous objectives, partial participation has effect proportional to rate of participation

- Consistent with $\sigma_{\text{fed}}^2 = \frac{1}{KL}\sum_{k=1}^{K} \frac{\sigma^2}{E} = \frac{1}{L}\frac{\sigma^2}{E}$.



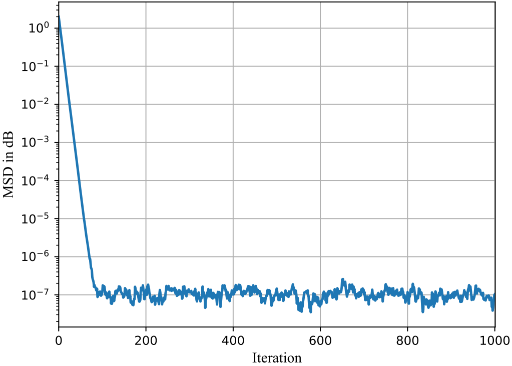
$K = L = 100,\ E = 1,\ \sigma_w^2 = 0$



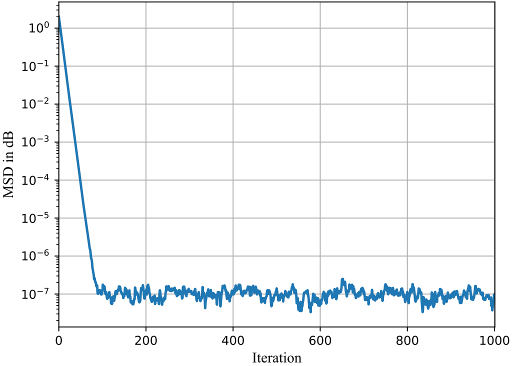
$K = 100,\ L = 10,\ E = 1,\ \sigma_w^2 = 0$

Observation #3: For homogeneous objectives, local updates benefit proportional to the number of local updates

- Consistent with $\sigma_{\text{fed}}^2 = \frac{1}{KL} \sum_{k=1}^{K} \frac{\sigma^2}{E} = \frac{1}{L}\frac{\sigma^2}{E}$.



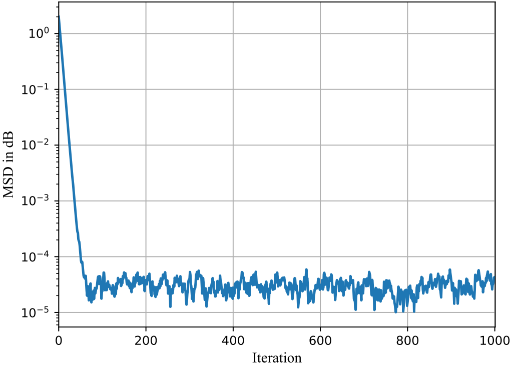
$$K = L = 100,\ E = 1,\ \sigma_w^2 = 0$$



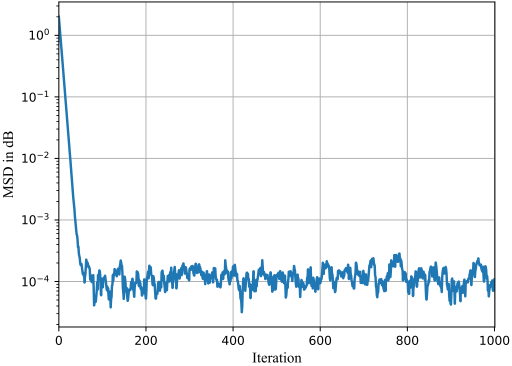
$$K = 100,\ L = 10,\ E = 10,\ \sigma_w^2 = 0$$

Observation #4: For heterogeneous objectives, the benefit of local updates does not outweigh the cost of partial participation

- $\sigma_{\text{fed}}^2 = \frac{1}{KL}\sum_{k=1}^{K}\left(2\frac{\beta_k^2}{E_k}\|w_k^o - w^o\|^2 + \frac{\sigma_k^2}{E_k}\right) + \frac{1}{KL}\frac{K-L}{K-1}\sum_{k=1}^{K}4\delta_k^2\|w_k^o - w^o\|^2.$



$$K = L = 100,\ E = 1,\ \sigma_w^2 = 0.1$$



$$K = 100,\ L = 10,\ E = 10,\ \sigma_w^2 = 0.1$$

Conclusion and Outlook

- We have seen how multiple agents can utilize a fusion center to collaborate in solving learning problems. This gave rise to a number of structures:
  - Centralized (full and regular participation)
  - Asynchronous (agents are sampled)
  - Federated (agents are sampled, local updates)
- In the best (i.e., homogeneous) case, this results in performance gains proportional to the number of participating agents, and the number of local updates.
- Heterogeneity deteriorates performance, but can be tackled via variance reduction.
- From tomorrow, we will focus on decentralized learning where agents rely on peer-to-peer interactions over a graph.

References and Further Reading

- General references and surveys:
  - 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.
  - S. Vlaski, S. Kar, A. H. Sayed and J. M. F. Moura, “Networked Signal and Information Processing: Learning by multiagent systems,” in *IEEE Signal Processing Magazine*, vol. 40, no. 5, pp. 92-105, July 2023,
- Performance guarantees as shown here:
  - E. Rizk, S. Vlaski and A. H. Sayed, “Federated Learning Under Importance Sampling,” in *IEEE Transactions on Signal Processing*, vol. 70, pp. 5381-5396, 2022.
- Variance-reduction for federated learning:
  - S.P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich and A. T. Suresh, “SCAFFOLD: Stochastic Controlled Averaging for Federated Learning”, in *Proceedings of the 37th International Conference on Machine Learning*, 2020.