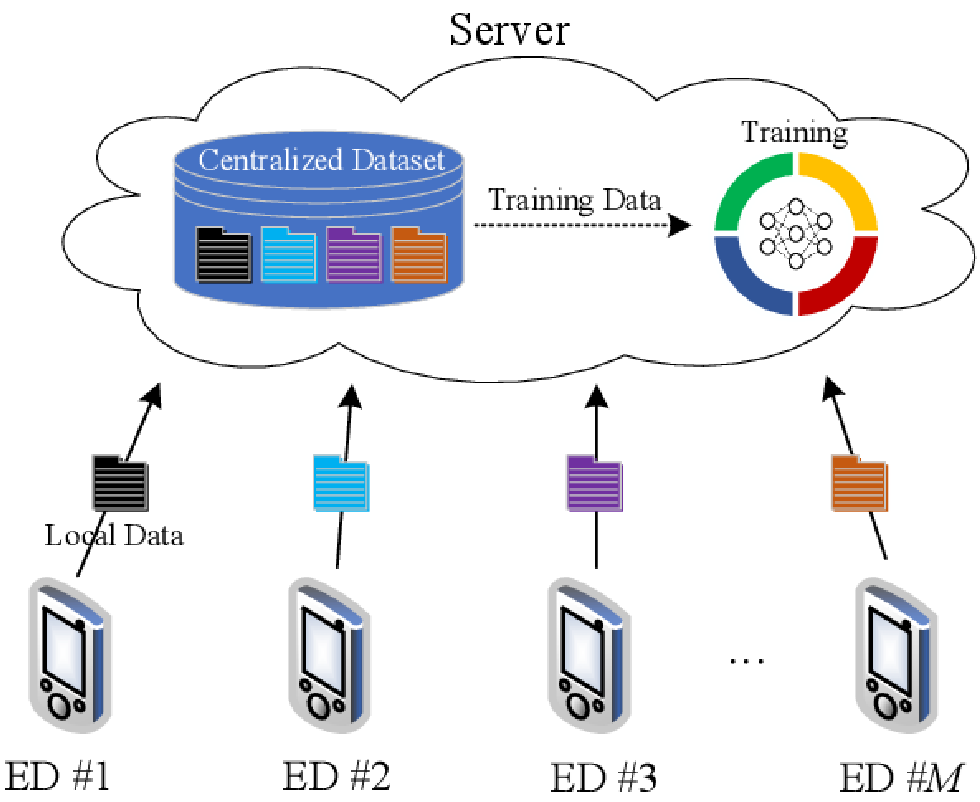
Dynamic federated learning

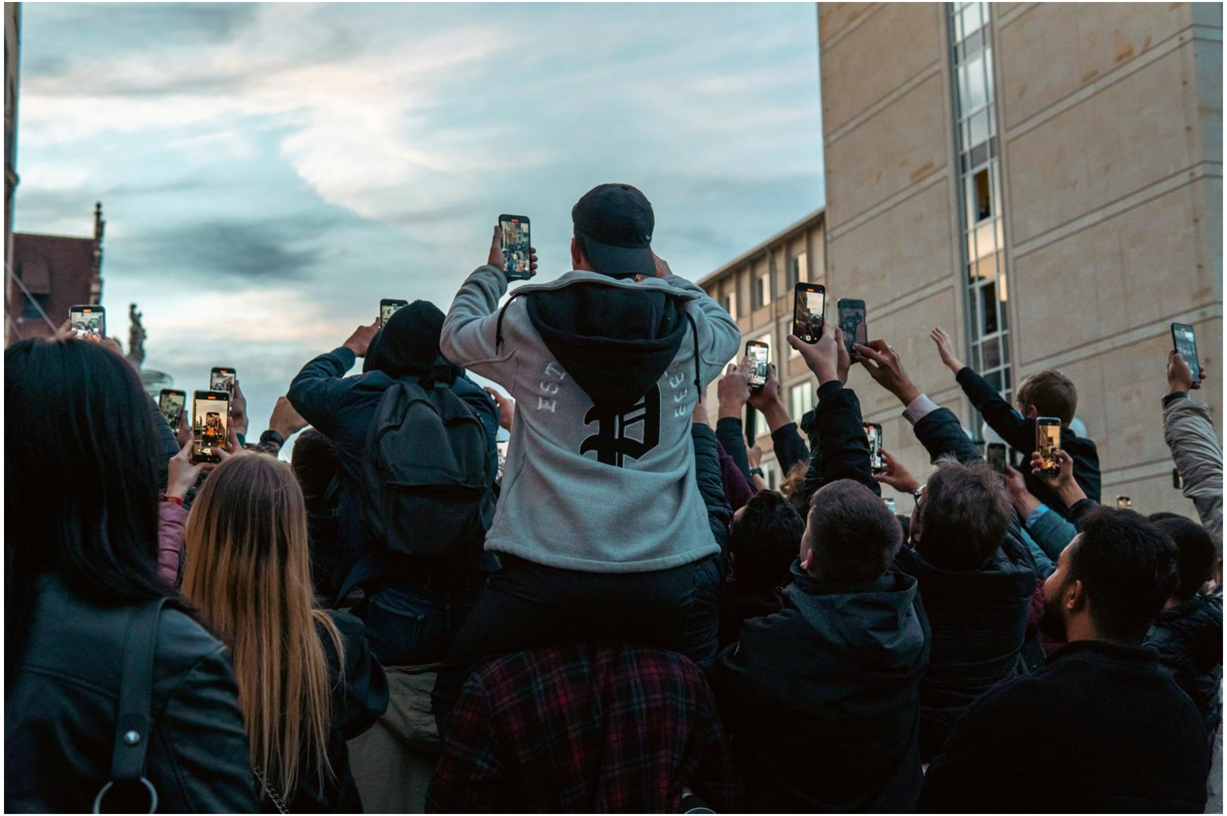
Elsa Rizk, Stefan Vlaski and Ali S. Hayed

Ilija Doknić and Stefan Komarica

Centralized learning

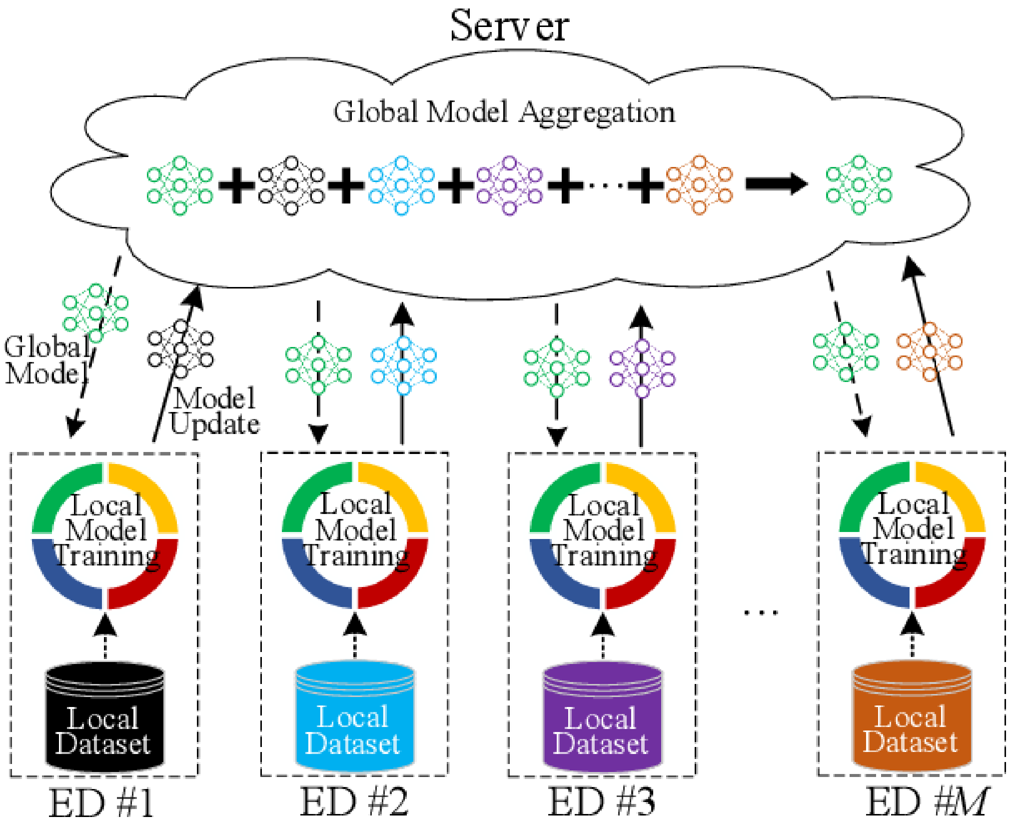


Dynamic federated learning



Source: Photo by Victoria Prymak on Unsplash

Federated learning



Federated vs Distributed learning

Differences:

- Connection
- Heterogeneity

Federated Averaging

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
  initialize w_0
  for each round $t = 1, 2, \dots$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of m clients)
    for each client $k \in S_t$ **in parallel do**
      $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
    $m_t \leftarrow \sum_{k \in S_t} n_k$
    $w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{m_t} w_{t+1}^k$ *// Erratum*[4]

ClientUpdate(k, w): *// Run on client k*
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

Dynamic Federated Averaging

Algorithm 2 (Dynamic Federated Averaging)

initialize w_0
for each iteration $i = 1, 2, \cdots$ **do**
  Select set of participating agents $\mathcal{L}_i$ by sampling L times from $\{1, \ldots, K\}$ without replacement.
  for each agent $k \in \mathcal{L}_i$ **do**
    initialize $\boldsymbol{w}_{k,-1} = \boldsymbol{w}_{i-1}$
    for each epoch $e = 1, 2, \cdots E_k$ **do**
      Find indices of the mini-batch sample $\mathcal{B}_{k,e}$ by sampling B_k times from $\{1, \ldots, N_k\}$ without replacement.
      $\boldsymbol{g} = \frac{1}{B_k} \sum_{b \in \mathcal{B}_{k,e}} \nabla_{w^\top} Q(\boldsymbol{w}_{k,e-1}; \boldsymbol{x}_{k,b})$
      $\boldsymbol{w}_{k,e} = \boldsymbol{w}_{k,e-1} - \mu \frac{1}{E_k} \boldsymbol{g}$
    end for
  end for
  $\boldsymbol{w}_i = \frac{1}{L} \sum_{k \in \mathcal{L}_i} \boldsymbol{w}_{k,E_k}$
end for

Convergence analysis

Assumptions:

- All of the risk functions are strongly convex and loss functions are convex. Both of them also have δ-Lipschitz gradients
- True model follows a Brownian random walk model
- For every instance of each local model it’s distance to the global model is bounded

Note regarding the second assumption

$$\boldsymbol{w}_i^o = \boldsymbol{w}_{i-1}^o + \boldsymbol{q}_i$$

$$\mathbb{E}\|\boldsymbol{q}_i\|^2 = \sigma_q^2$$

σ_q^2 - Drift parameter

Error Recursion

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \frac{1}{L} \sum_{\ell \in \mathcal{L}_i} \frac{1}{E_\ell B_\ell} \sum_{e=0}^{E_\ell - 1} \sum_{b \in \mathcal{B}_{\ell,e}} \nabla_{w^\top} Q_\ell(\boldsymbol{w}_{\ell,e-1}; \boldsymbol{x}_{\ell,b})$$

$$= \boldsymbol{w}_{i-1} - \mu \frac{1}{K} \sum_{k=1}^{K} \nabla_{w^\top} P_k(\boldsymbol{w}_{i-1}) - \mu \boldsymbol{s}_i - \mu \boldsymbol{d}_i,$$

$\boldsymbol{s}_i$ - Results from stochastic approximation (The gradient noise)

$\boldsymbol{d}_i$ - Results from the incremental implementation

$$\mathbb{E}\,\{\boldsymbol{s}_i|\boldsymbol{w}_{i-1}\} = 0,$$

$$\mathbb{E}\left\{\|\boldsymbol{s}_i\|^2|\boldsymbol{w}_{i-1}\right\} \leq \beta_s^2\mathbb{E}\|\boldsymbol{w}_{i-1}^o - \boldsymbol{w}_{i-1}\|^2 + \sigma_s^2 + \epsilon^2$$

$$\beta_s^2 \triangleq \frac{1}{KL}\sum_{k=1}^{K}\Big(6\tau_{s,k} + 2\tau_\epsilon\Big)\delta^2,$$

Average data variability

$$\sigma_s^2 \triangleq \frac{3}{KL}\sum_{k=1}^{K}\tau_{s,k}\mathbb{E}\|\nabla_{w^\top}Q_k(\boldsymbol{w}_{i-1}^o;\boldsymbol{x}_k) - \nabla_{w^\top}P_k(\boldsymbol{w}_{i-1}^o)\|^2,$$

$$\epsilon^2 \triangleq \frac{2}{KL}\sum_{k=1}^{K}\tau_\epsilon^2\mathbb{E}\|\nabla_{w^\top}P_k(\boldsymbol{w}_{i-1}^o)\|^2,$$

Model variability

$$\tau_{s,k} \triangleq \frac{N_k - B_k}{(N_k-1)B_kE_k}, \quad \tau_\epsilon \triangleq \frac{K-L}{K-1}.$$

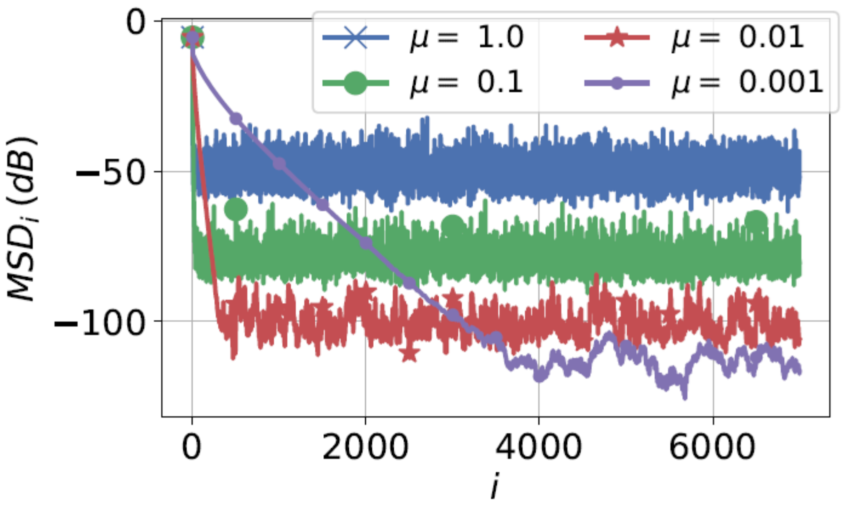
Experimental analysis

- We examine use of different hyperparameters one the behaviour of our algorithm
- We validate experimental results by changing the step size
- We plot the averaged mean-squared-deviation (MSD) curves in log domain (values in dB)

Experimental setup

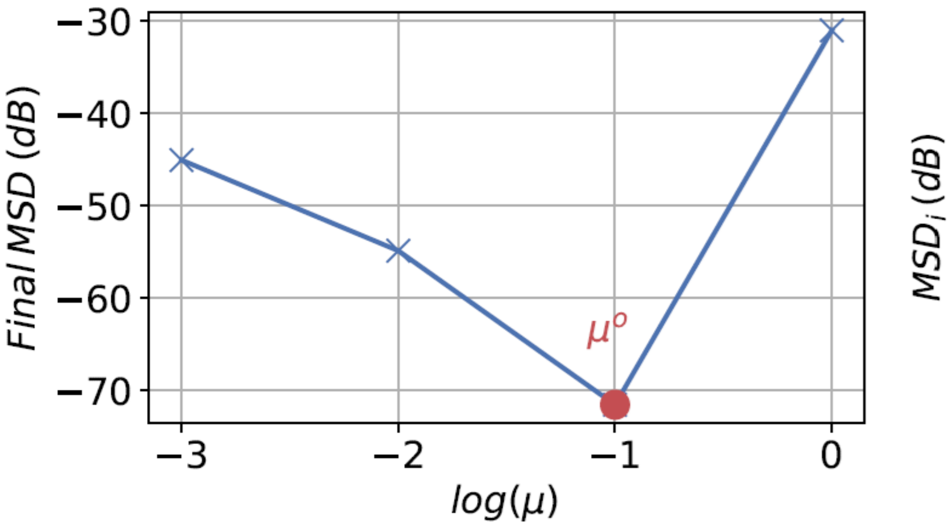
- Start at random value for model parameter
- Model the change across the true parameters by adding randomly sampled constants
- We assume we have K = 20 agents, with L = 7 active agents and we set batch sizes and epoch sizes to different values

$$\boldsymbol{c}_k \sim \mathcal{N}(0, \sigma_c^2) \qquad \boldsymbol{w}_{k,i}^{\star} = \boldsymbol{w}_i^{\star} + \boldsymbol{c}_k$$



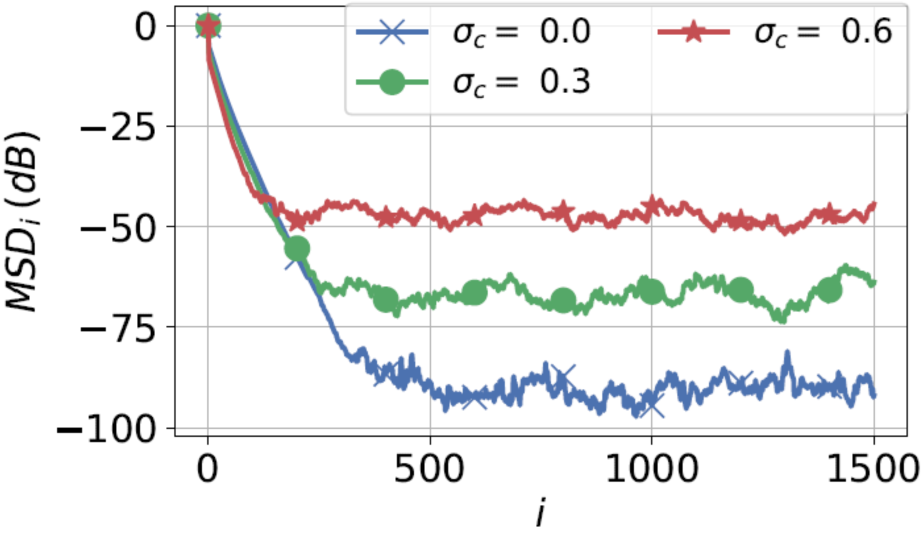
Stationary case: varying μ

$$\sigma_c^2 = 0.1 \qquad \sigma_q^2 = 0$$



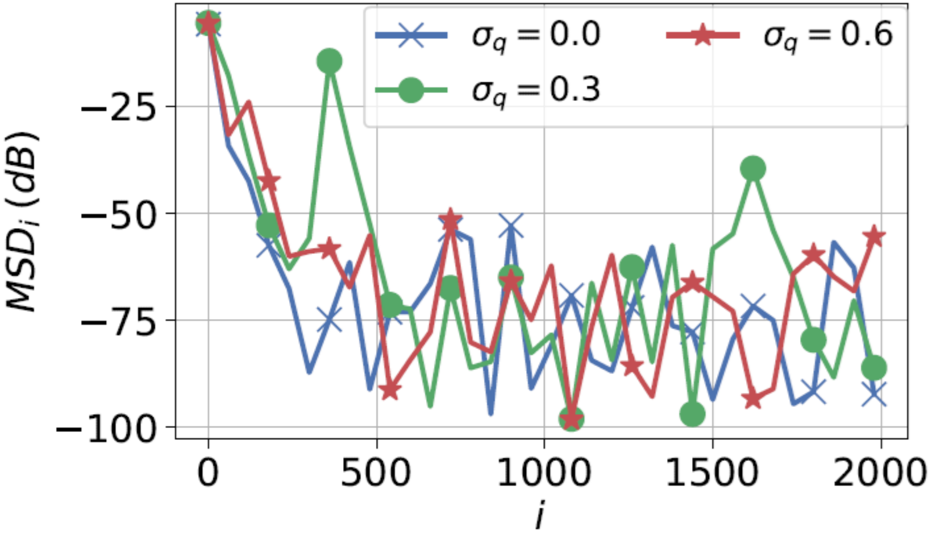
Non-stationary case: varying μ

$$\sigma_c^2 = 0.1 \qquad \sigma_q^2 = 0.01$$



Varying σ_c^2

$\sigma_q^2 = 0 \qquad \mu = 0.01$



Varying σ_q^2

$\sigma_c^2 = 0.1 \qquad \mu = 0.01$

Conclusion

The authors based their work on developing a convergence analysis on the modified version the FedAvg algorithm. They were able to identify the most important components that affect performance:

- Step size
- Agent heterogeneity
- Drift variance

Questions?