article amsmath amsfonts amssymb geometry hyperref margin=1in

# FLeNS: Federated Learning with Enhanced Nesterov-Newton Sketch

Your Name

September 12, 2024

## 1 Introduction

FLeNS is designed to optimize a global model by combining Nesterov's acceleration with a sketched Hessian matrix in a federated learning setting. This approach leverages the momentum from Nesterov's method to accelerate convergence while reducing the dimensionality of the Hessian matrix through sketching, making the optimization process more efficient.

article algorithm algpseudocode amsmath amsfonts

## 2 Computational Complexity

### 2.1 Local Computation

**Gradient Calculation:** Each client $j$ computes the local gradient $g_{D_j,t}$ at iteration $t$:

$$g_{D_j,t} = \frac{1}{n_j} \sum_{i=1}^{n_j} \nabla_w \ell(f(w_t; x_{ij}), y_{ij}) + \lambda w_t \tag{1}$$

The time complexity for computing the local gradient is:

$$O(n_j M) \tag{2}$$

where $n_j$ is the number of data points on client $j$, and $M$ is the dimension of the model parameters.

**Hessian Sketching:** Each client applies a sketching operation to the Hessian matrix using a sketch matrix $S_j \in \mathbb{R}^{k \times d}$:

$$\tilde{H}_{D_j,t} = S_j^\top \nabla^2 L_{D_j,t} S_j \tag{3}$$

The time complexity for computing the sketched Hessian is:

$$O(n_j M^2) \text{ (for the full Hessian)} \rightarrow O(n_j k M) \text{ (after sketching)} \tag{4}$$

where $k$ is the sketch size, typically $k \ll M$.

**Nesterov's Acceleration:** Nesterov's accelerated gradient step involves updating the model with momentum:

$$v_t = w_t + \beta_t (w_t - w_{t-1}) \tag{5}$$

The time complexity for this vector update is:

$$O(M) \tag{6}$$

**Total Local Computation:** The total computational cost per client for each iteration, considering gradient computation, Hessian sketching, and Nesterov's acceleration, is:

$$O(n_j M + n_j k M + M) \tag{7}$$

Simplified, the dominant term is:

$$O(n_j k M) \tag{8}$$

### 2.2 Global Computation

**Aggregation:** The global server aggregates the sketched Hessians and gradients from all clients:

$$\tilde{H}_{D,t} = \sum_{j=1}^{m} \frac{n_j}{N} \tilde{H}_{D_j,t} \tag{9}$$

$$g_{D,t} = \sum_{j=1}^{m} \frac{n_j}{N} g_{D_j,t} \tag{10}$$

The time complexity for aggregating the sketched Hessians and gradients across $m$ clients is:

$$O(mkM) \tag{11}$$

**Global Model Update:** The global model is updated using the Nesterov-accelerated gradient and the sketched Hessian:

$$w_{t+1} = v_t - \mu \tilde{H}_{D,t}^{-1} g_{D,t} \tag{12}$$

The time complexity for inverting the aggregated sketched Hessian is:

$$O(k^2M + kM^2) \text{ (for inversion)} \tag{13}$$

$$O(kM^2) \text{ (dominant term)} \tag{14}$$

**Total Global Computation:** The overall global computation per iteration is:

$$O(mkM + kM^2) \tag{15}$$

# 3 Communication Complexity

## 3.1 Local to Global Communication

**Gradient and Hessian Communication:** Each client sends its local sketched Hessian $\tilde{H}_{D_j,t}$ (of size $k \times M$) and gradient $g_{D_j,t}$ (of size $M$) to the global server.

$$O(kM + M) \approx O(kM) \tag{16}$$

Total communication across all clients is:

$$O(mkM) \tag{17}$$

## 3.2 Global to Local Communication

**Model Update Communication:** The global server sends the updated model $w_{t+1}$ back to each client.

$$O(M) \tag{18}$$

Total communication across all clients is:

$$O(mM) \tag{19}$$

# 4 Overall Complexity Analysis

## 4.1 Computational Complexity

**Total Local Computation:** The overall local computation per client per iteration is:

$$O(n_jkM) \tag{20}$$

**Total Global Computation:** The overall global computation per iteration is:

$$O(mkM + kM^2) \tag{21}$$

## 4.2 Communication Complexity

**Total Communication (Local to Global + Global to Local):** The overall communication complexity per iteration is:

$$O(mkM + mM) \approx O(mkM) \tag{22}$$

# 5 Comparison to Other Methods

**Efficiency:** FLeNS improves efficiency by leveraging the momentum from Nesterov's acceleration, which can lead to faster convergence compared to standard Newton's methods. The use of a sketched Hessian matrix reduces the dimensionality of the problem, significantly lowering both the computational and communication costs.

**Scalability:** The scalability of FLeNS is well-suited for large-scale federated learning settings where communication costs are critical. By sketching the Hessian, the size of the data communicated between clients and the global server is reduced, allowing the algorithm to handle large models and datasets more efficiently.

**Convergence:** The convergence speed is enhanced by Nesterov's acceleration, which provides an additional boost over standard gradient descent methods. The use of sketching, while reducing the complexity, does not significantly impact the convergence quality due to the retention of key Hessian information through the sketching process.

# 6 Summary

| Aspect | FLeNS |
|---|---|
| Local Computation | $O(n_j kM)$ |
| Global Computation | $O(mkM + kM^2)$ |
| Communication (Local-Global) | $O(mkM)$ |
| Communication (Global-Local) | $O(mM)$ |
| Total Complexity | Lower than full Hessian methods, efficient due to sketching and acceleration |
| Convergence Speed | Fast due to Nesterov's acceleration, with good scalability |

article amsmath amssymb amsfonts geometry hyperref margin=1in

Convergence Analysis of FLeNS: Federated Learning with Enhanced Nesterov-Newton Sketch Your Name September 12, 2024

# 7 Overview

FLeNS is designed to optimize a global model by leveraging both the acceleration properties of Nesterov's method and the dimensionality reduction provided by Hessian sketching. The key idea is to achieve faster convergence than traditional methods by incorporating momentum while also reducing the computational and communication burden via sketching.

article amsmath amssymb amsfonts geometry algorithm algorithmic margin=1in

Convergence Analysis for FLeNS: Federated Learning with Enhanced Nesterov-Newton Sketch Your Name September 12, 2024

# 8 Introduction

FLeNS (Federated Learning with Enhanced Nesterov-Newton Sketch) extends Newton's method by incorporating Nesterov's acceleration and Hessian sketching in a federated learning setting. This document presents the convergence analysis of the FLeNS algorithm, highlighting how the combination of these techniques affects the convergence rate.

# 9 Algorithm Overview

**Objective:** Optimize a global model using Nesterov's acceleration combined with sketching of the Hessian matrix in a federated learning setting.

**Global Objective Function:**

$$w^* = \arg\min_{w\in\mathbb{R}^d} \left\{ L(w) = \frac{1}{N}\sum_{j=1}^{m}\sum_{i=1}^{n_j} \ell(f(w;x_{ij}),y_{ij}) + \frac{\lambda}{2}\|w\|^2 \right\}$$

where $N = \sum_{j=1}^{m} n_j$ is the total number of data points across all clients.

# 10 Convergence Analysis

## 10.1 Nesterov's Acceleration

In FLeNS, the weight update rule incorporates Nesterov's acceleration:

$$v_t = w_t + \beta_t(w_t - w_{t-1})$$

Here, $v_t$ is the accelerated variable, and $\beta_t$ is the momentum term. The gradient $g_{D_j,t+1}$ and the sketched Hessian $\tilde{H}_{D_j,t+1}$ are computed based on $v_t$ rather than $w_t$.

## 10.2 Local Computation with Sketching

Each client $j$ computes the local gradient and applies sketching to the Hessian matrix:

$$g_{D_j,t} = \frac{1}{n_j}\sum_{i=1}^{n_j} \nabla_w \ell(f(v_t;x_{ij}),y_{ij}) + \lambda v_t$$

$$\tilde{H}_{D_j,t} = S_j^\top \nabla^2 L_{D_j,t} S_j$$

## 10.3 Global Aggregation and Update

The global server aggregates the sketched Hessians and gradients from all clients:

$$\tilde{H}_{D,t} = \sum_{j=1}^{m} \frac{n_j}{N} \tilde{H}_{D_j,t+1}$$

$$g_{D,t} = \sum_{j=1}^{m} \frac{n_j}{N} g_{D_j,t+1}$$

The global update rule is then applied using the accelerated variable:

$$w_{t+1} = v_t - \mu \tilde{H}_{D,t}^{-1} g_{D,t}$$

### 10.4 Error Bound and Convergence Rate

The error bound in FLeNS is influenced by both the Hessian sketching and the momentum term from Nesterov's acceleration:

$$\|w_{t+1} - w^*\|^2 \leq \epsilon \left( \frac{\beta}{\gamma} \|v_t - w^*\|^2 + \frac{4L}{\gamma} \|v_t - w^*\|^4 \right)$$

where $\epsilon$ is the tolerance, $\gamma$ is the curvature constant, and $L$ is the Lipschitz constant. The sketch size $k$ must satisfy:

$$m \gtrsim \epsilon^{-2} M = \frac{1}{\log^2(1+t)} M$$

to ensure that the approximation error remains within acceptable bounds.

### 10.5 Final Convergence Guarantee

Under appropriate conditions, FLeNS achieves super-linear convergence:

$$\|w_{t+1} - w^*\|^2 \leq \frac{1}{\log(1+t)} \left( \frac{\beta}{\gamma} \|w_t - w^*\|^2 + \frac{4L}{\gamma} \|w_t - w^*\|^4 \right)$$

The use of Nesterov's acceleration leads to faster error reduction compared to traditional Newton's method.

### 10.6 Practical Considerations

**Initialization:** The initial point $w_0$ should be close to the optimal solution $w^*$ to achieve super-linear convergence.

**Communication Complexity:** The communication cost per iteration depends on the sketch size $k$ and the number of clients $m$, similar to FedNS but with the added complexity of Nesterov's acceleration.

## 11 Conclusion

The convergence analysis for FLeNS shows that by combining Nesterov's acceleration with Hessian sketching, the algorithm can achieve super-linear convergence rates under the right conditions. Proper parameter tuning is crucial for ensuring that FLeNS performs optimally in federated learning environments.

article amsmath amssymb amsfonts geometry margin=1in

Generalization Analysis for FLeNS: Federated Learning with Enhanced Nesterov-Newton Sketch Your Name September 12, 2024

## 12 Introduction

FLeNS (Federated Learning with Enhanced Nesterov-Newton Sketch) is a federated learning algorithm that combines Nesterov's acceleration with Hessian sketching to optimize a global model. This document presents the generalization analysis for FLeNS, specifically focusing on how these techniques affect the generalization error.

## 13 General Setup

**Objective:** The global objective function is defined as:

$$w^* = \arg\min_{w \in \mathbb{R}^d} \left\{ L(w) = \frac{1}{N} \sum_{j=1}^{m} \sum_{i=1}^{n_j} \ell(f(w; x_{ij}), y_{ij}) + \frac{\lambda}{2} \|w\|^2 \right\},$$

where $N = \sum_{j=1}^{m} n_j$ is the total number of data points across all clients.

## 14 Assumptions

- **Lipschitz Continuity:** The loss function $\ell(f(w; x_{ij}), y_{ij})$ is Lipschitz continuous with constant $L$. This ensures stability in the gradient computation.

- **Strong Convexity:** The loss function is strongly convex with constant $\lambda > 0$, implying a well-defined global minimum.
- **Bounded Gradient Norm:** The norm of the gradient $\|\nabla_w \ell(f(w; x_{ij}), y_{ij})\|$ is uniformly bounded across all data points and clients.

# 15 Generalization Error Bound

## 15.1 Federated Learning Setting

In FLeNS, each client computes a local gradient and a sketched Hessian, which are aggregated at the global server. The global model update incorporates Nesterov's acceleration to enhance convergence.

## 15.2 Generalization Error Definition

The generalization error is defined as the difference between the expected loss of the learned model $w_T$ after $T$ iterations and the loss of the optimal model $w^*$:

$$\mathbb{E}[L(w_T)] - L(w^*),$$

where $\mathbb{E}[\cdot]$ denotes the expectation over the distribution of the data.

## 15.3 Error Bound

Under the Lipschitz continuity and strong convexity assumptions, the generalization error after $T$ iterations can be bounded as:

$$\mathbb{E}[L(w_T)] - L(w^*) \leq O\left(\frac{L\|w_0 - w^*\|^2}{T^2}\right) + O\left(\frac{1}{\sqrt{k}}\right),$$

where $k$ is the sketch size, $L$ is the Lipschitz constant, and $\|w_0 - w^*\|^2$ is the initial error.

- The first term $O\left(\frac{L\|w_0-w^*\|^2}{T^2}\right)$ represents the rapid convergence due to Nesterov's acceleration.
- The second term $O\left(\frac{1}{\sqrt{k}}\right)$ accounts for the error introduced by Hessian sketching. As $k$ increases, this error decreases, improving generalization.

## 15.4 Sketch Size and Generalization

The sketch size $k$ must be large enough to ensure accurate Hessian approximation. A typical choice is:

$$k = \Omega(\sqrt{d}),$$

where $d$ is the dimensionality of the model.

# 16 Impact of Nesterov's Acceleration

Nesterov's acceleration not only speeds up convergence but also helps in reducing the generalization error by allowing the model to reach lower optimization errors faster. This results in a lower generalization error within the same number of iterations compared to methods without acceleration.

# 17 Conclusion

The generalization analysis of FLeNS shows that the combination of Nesterov's acceleration and Hessian sketching can effectively balance fast convergence with low generalization error. By carefully choosing the sketch size $k$ and leveraging the momentum term from Nesterov's method, FLeNS achieves efficient and scalable performance in federated learning environments.

**Algorithm 1** FLeNS: Federated Learning with Enhanced Nesterov-Newton Sketch

**Objective:** Optimize a global model using Nesterov's acceleration combined with sketching of the Hessian matrix in a federated learning setting.

**Setup:** Let $D_j$ represent the local dataset at client $j$, where $j = 1, \dots, m$. The global objective function is:

$$w^* = \arg\min_{w \in \mathbb{R}^d} \left\{ L(w) = \frac{1}{N} \sum_{j=1}^{m} \sum_{i=1}^{n_j} \ell(f(w; x_{ij}), y_{ij}) + \frac{\lambda}{2} \|w\|^2 \right\}$$

where $N = \sum_{j=1}^{m} n_j$ is the total number of data points across all clients.

```
Step 1: Local Gradient Computation and Sketching
for each client $j \in \{1, \dots, m\}$ do
    Compute the local gradient $g_{D_j,t}$:
        $g_{D_j,t} = \frac{1}{n_j} \sum_{i=1}^{n_j} \nabla_w \ell(f(w_t; x_{ij}), y_{ij}) + \lambda w_t$
    Apply sketching to the Hessian's feature dimension using sketch matrix $S_j \in \mathbb{R}^{k \times d}$:
        $\tilde{H}_{D_j,t} = S_j^\top \nabla^2 L_{D_j,t} S_j$
end for
Step 2: Nesterov's Acceleration
for each client $j$ do
    Apply Nesterov's acceleration:
        $v_t = w_t + \beta_t (w_t - w_{t-1})$
    Update the gradient and sketched Hessian based on $v_t$:
        $g_{D_j,t+1} = \nabla L(v_t), \quad \tilde{H}_{D_j,t+1} = \nabla^2 L(v_t)$
end for
Step 3: Communication to Global Server
for each client $j$ do
    Send $\tilde{H}_{D_j,t+1}$ and $g_{D_j,t+1}$ to the global server.
end for
Step 4: Aggregation at Global Server
Aggregate the sketched Hessians and gradients:
    $\tilde{H}_{D,t} = \sum_{j=1}^{m} \frac{n_j}{N} \tilde{H}_{D_j,t+1}$
    $g_{D,t} = \sum_{j=1}^{m} \frac{n_j}{N} g_{D_j,t+1}$
Step 5: Global Model Update
Update the global model parameters:
    $w_{t+1} = w_t - \mu \tilde{H}_{D,t}^{-1} g_{D,t}$
where $\mu$ is the step size.
Step 6: Iterate
Repeat steps 1-5 until convergence criteria are met.
```