FedND: Federated Newton Direction for Federated Learning Environment

Samir Aliyev
General and Applied Mathematics
Azerbaijan State Oil and Industry University
Baku, Azerbaijan
aliyev.samir@asoiu.edu.az

***Abstract*—Federated Learning (FL) is a decentralized approach to machine learning that allows models to be trained across multiple clients without sharing raw data. This study investigates the performance of two federated optimization algorithms—Federated Newton Direction (FedND) and Federated Stochastic Gradient Descent (FedSGD)—under different client weighting schemes. Two weighting strategies were applied: the conventional sample-size-based weighting and a Fuzzy Inference System (FIS)-based approach. The results indicate that FedND consistently outperformed FedSGD in terms of accuracy, AUC, and F1 Score. Furthermore, FIS-based weighting provided additional improvements, particularly in accuracy and F1 Score, by addressing client heterogeneity more effectively. While FedND achieved superior overall performance, the study highlights that adaptive weighting methods can further enhance federated learning efficiency. These findings contribute to the development of more robust aggregation techniques for FL frameworks, particularly in heterogeneous environments.**

Keywords—Federated Learning, Federated Stochastic Gradient, Second-Order methods, Newton Method, Fuzzy Inference System,

I. Introduction

As the popularity of Machine Learning increase in several fields science and industry, several challenges emerge with it. Traditional machine learning algorithms is centralized. Data is collected from multiple sources and used to train the model which is used by clients. Nowadays, thanks to IoT devices, the amount of data is increasing like it never does before [1]. The problem however, lies on the issue of preserving data privacy. Sharing data with server opens several privacy issues which may not be acceptable for data sources considering the price of data. This problem is tackled by the previously emerged ML approach called Federated Learning [2]. Federated learning is distributed machine learning approach where clients global model is emerged without direct access to data. IoT devices have considerable computational capabilities which allow them calculate model updates on their own. By doing so, clients can participate on the training of global model which they will use, without sending data thus preserving the privacy. FL has been proven to be effective in several fields such as healthcare, autonomous cars.

However, FL comes with several challenges. One of these challenges is the aggregation process of local models [3]. This step is crucial because the choice of the wrong aggregation algorithm may lead to suboptimal results. These algorithms differ from each other based on their communication cost, vulnerability, averaging and etc. One of these algorithms is called Federated Stochastic Gradient Descent (FedSGD) where clients calculate gradients and send them to server to be aggregated. Obtained gradient then, is used to make update. Aggregation is done by weighted averaging where weights are determined by number of data samples used during calculation of gradients. In this work, we compared the performance of FedSGD with another algorithm called Federated Newton Direction (FedND) by calculating not only gradients, but also hessian which includes second order derivatives. But instead of sending them to server for aggregation we find direction by multiplying inverse of hessian with gradient to obtain direction. We also tested how the performance will change if use different weighting approach which has been tested in one of our previous works.

Second section of this paper is dedicated to related works done before. In the third section, we discussed the main methods, give theoretical background about the tools we have used to reach the goal. Results section demonstrates the experimental setup, dataset and obtained observations during the experiments. Finally, we discussed the advantages and disadvantages and possible future direction for improvement.

II. Related Works

FedSGD is one the fundamental algorithms used in FL environment [4]. One of the main advantages of this algorithm is its simplicity and speed. It only uses gradient of the function for aggregation which makes it cost effective. However, its main disadvantage is that it each client participates in communication round must wait until all gradients are gathered for update. Federated Averaging (FedAVG) algorithm overcomes this difficulty by making local model updated and these parameters are aggregated into one global model [5]. This approach allows clients to make several updates before sending model to server for aggregation. In both cases, aggregation is done by weighted averaging. Those weights are determined by number of training samples they used to train their model. There have been several studies to tackles this naïve approach. One of them is FedProx which also uses the number of local updates in one client for making decision [6]. Soft Computing approaches such as AHP, FIS also have been applied successfully in FL settings to improve the performance of the model [7][8].

Second order methods have also been applied in federated learning settings [9]. However, because it requires the hessian, it makes communications more costly. Newton method which applied uses inverse of hessian and gradient is the most popular second order method. Newton method converges much faster since it uses quadratic approximation function but it does not necessarily mean less communication cost. Sending hessian is more expensive. FedNew method was proposed to overcome this challenge. This algorithm reduces the communication cost and additionally preserving data privacy by estimating hessian-gradient-product in each iteration [10]. By not sending gradients, this method makes it less vulnerable to gradient conversion and data reconstruction attacks [11]. In another work, authors implemented Quasi-

Newton method which instead of calculating directly inverse of hessian, it finds approximation [12].

In this work we tried implement method in which hessian-gradient-product is calculated at server and obtain direction is sent to server. By doing so, gradients are also hided which makes it more secure. We also used FIS to compare the performances of different client weighting schemes.

III. Methodology

This section is dedicated to the method and tools used in this research. Theoretical information behind the methods and our influence is discussed in coming subsections.

A. Federated Stochastic Gradients

Federated Stochastic Gradient Descent (FedSGD) is naïve approach used as baseline method in many federated learning aggregation method. Idea is very similar to mini-batch stochastic gradient descent in which random batch of samples is selected and gradient is calculated based on that sample. In Federated Learning environment, each client is treated as batch. Clients calculate their gradient and send them to be aggregated in the server. Aggregated gradients are used to update the model to be used in clients. Mathematically in can be expressed as:

$$w_{i+1} = w_i - \alpha \sum_{k=1}^{K} \frac{n_k}{n} g_k$$

Where, w_i is the parameters in iteration I, n_k is the training size of client k, g_k is the gradients calculated in client k, and α is the learning rate. We have to take this into consideration that each client is given weight based on training samples used to obtain gradient which may be suboptimal. We tried to overcome this difficulty by using Fuzzy Inference System which will be discussed in following sections.

B. Federated Newton Direction

FedSGD uses only first derivatives of the function to update the parameters of the model. Newton method is second-order method which uses hessian of the function alongside the gradient.

$$w_{i+1} = w_i - H^{-1} g$$

In this work, instead of sending hessian and gradient to server to be aggregated, we multiplied the direction by multiplying them. Obtained directions were averaged and used to update the model. By doing so, we reduce the communication cost by not sending hessian matrix which can be enormous with big models.

$$For\ each\ client\ k, S_k = H_k^{-1} g_k$$

$$w_{i+1} = w_i - \sum_{k=1}^{K} \frac{n_k}{n} S_k$$

As a baseline, we gave clients weights as it is in original FedSGD. Additionally, weights obtained by FIS also experimented and compared with the other methods.

C. Fuzzy Inference System

Fuzzy Inference System is a soft computing approach which is used make decision in backing on fuzzy logic. It is easily understandable by human because of If-Then rules. These rules are designed by experts in the specific area and aggregated AND/OR operators. Here we created Mamdani-type FIS with two input variable which is size and the balance of classes in the dataset. Output is the given weights to each client. The distribution of classes in each client was calculated by GINI index given in equation x.

$$GINI = 1 - \sum_i p_i^2$$

Figure 1 represents the structure of FIS used in this work.

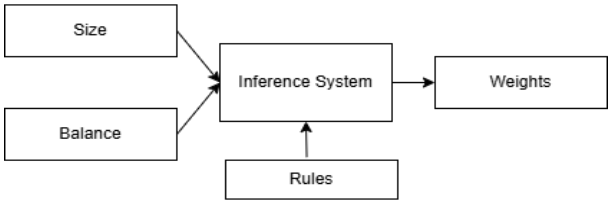

Figure 1. FIS structure

FIS system consists of three main stages. First step is fuzzification. Fuzzification is a mathematical process that transforms input values (from a defined range) into corresponding membership values within a fuzzy set. It takes inputs such as x and y and assigns them membership degrees μ(x) and μ(y), which always fall within the range [0, 1]. After fuzzification complete, fuzzy inputs are given to inference system. he fuzzy inference mechanism operates within a Fuzzy Inference System (FIS), which integrates multiple input variables into coherent logical rules to derive system outputs. In such systems, individual If-Then rules may incorporate two or more antecedent variables, necessitating the application of fuzzy operators (e.g., AND, OR) to resolve composite conditions. The inference process unfolds in three sequential stages:

- Application of Fuzzy Operators: Input membership degrees are combined using logical operations. For instance, the AND operator (modeled via a minimum function) or the OR operator (modeled via a maximum function) may be employed to evaluate rule antecedents.
- Implication: Each rule generates an output fuzzy set by truncating or scaling the consequent membership function. In this study, the minimum operator was utilized for implication, whereby the rule's antecedent strength (derived from inputs) directly bounds the consequent's membership function.
- Aggregation: The outputs of all activated rules are unified into a single composite fuzzy set. Here, the maximum method was adopted for aggregation, superimposing individual rule outputs to construct a holistic representation of the system's behavior.

Lastly. defuzzification stage converts the aggregated fuzzy set into a precise numerical value. This process uses methods like the Centroid of Area (COA), which calculates the weighted center of the aggregated fuzzy set. Specifically, the COA computes the center of each sub-region and sums their contributions to produce the final output. Mathematically, this is represented as:

$$COA = \frac{\int \mu(x) x dx}{\int \mu(x) dx}$$

Figure 2, demonstrates the visual representations of membership function. Rules used in during research is given in Table I.

IV. EXPERIMENTS AND RESULTS

A. Dataset

The experimental analysis utilized the MAGIC Gamma Telescope Dataset, comprising 19,020 entries with 11 attributes, categorized into two classes: gamma and hadron. The dataset exhibits a class imbalance, with 12,332 instances labeled as gamma and 6,682 as hadron. Notably, while hadron events are typically more prevalent in practice, a subset of these was retained for methodological considerations.

For the study, the dataset was partitioned into five client subsets through randomized allocation. The distribution strategy intentionally introduced heterogeneity in sample size across clients, resulting in varying numbers of instances per subset. Additionally, label distribution heterogeneity was incorporated: certain clients were assigned data with a near-balanced ratio of gamma-to-hadron samples, while others received skewed distributions to emulate real-world scenarios where class proportions may diverge significantly. This design ensured variability in both data volume and class representation across clients, reflecting diverse operational conditions.

B. Setup and Results

As a base model linear logistic regression was used in all clients. Algorithms were tested in four different setup: FedSGD with original weighting and FIS, and FedND with original weighting and FIS. Considering the class distribution besides accuracy, we also tested AUC (Area Under the Curve) score for validation. Table II demonstrates the observer results from experiments.

From the results, we can observe that FedND consistently outperformed FedSGD across all evaluation metrics. The average accuracy of FedND with original weights (0.778) and FIS weights (0.782) was significantly higher than FedSGD, which achieved 0.73 with original weights and 0.736 with FIS weights. Similarly, the F1 Score, which accounts for both precision and recall, showed an improvement in FedND (0.784 with original weights and 0.78 with FIS weights) compared to FedSGD (0.734 with original weights and 0.732 with FIS weights). The AUC Score, an indicator of the model's discriminatory power, also demonstrated a similar trend, with FedND achieving higher scores than FedSGD in both weight assignment methods. The introduction of FIS-based weighting resulted in minor improvements in most cases, particularly for FedND. For example, FedND with FIS weights improved the average accuracy from 0.778 to 0.782, while FedSGD showed a marginal increase from 0.73 to 0.736. However, in some instances, the AUC score slightly decreased when using FIS weights, suggesting that while the weighting strategy optimized accuracy and F1 Score, its impact on the model's ranking capability was less pronounced them in the server.

In FedSGD, Client 5 consistently achieved the highest accuracy and F1 Score, suggesting a more balanced or informative dataset, while Client 3 showed the lowest performance, likely due to data imbalance. FedND reduced these discrepancies, leading to more uniform accuracy and F1 Scores across clients.

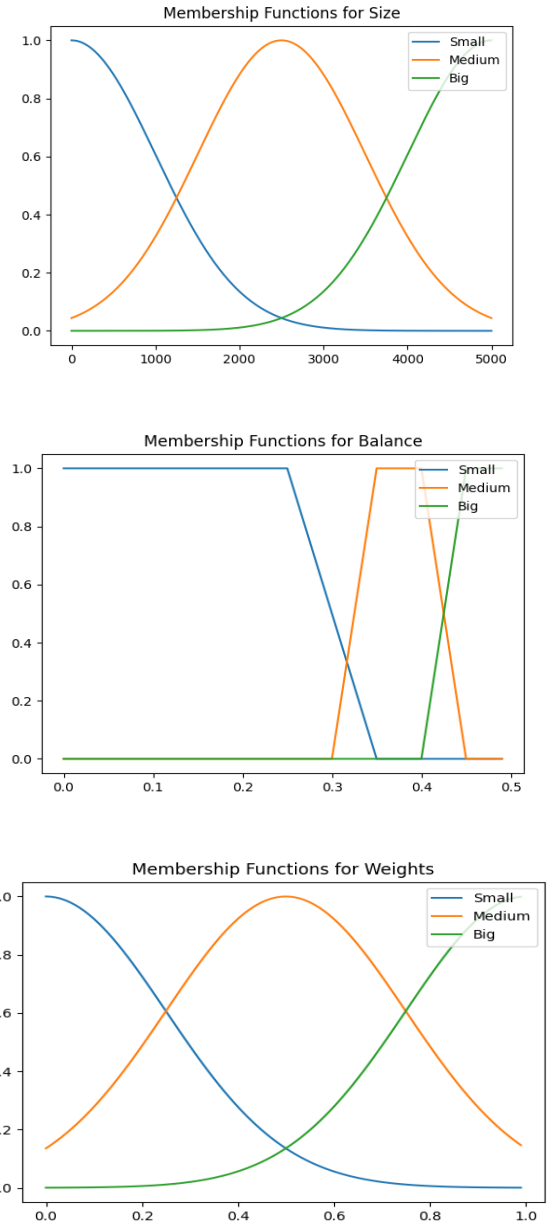

Figure 2 Membership Functions

TABLE I. IF-THEN RULES

Antecedent	Consequent
Size=*Small* and Balance= *Highly Imbalanced*	Weight= *Small*
Size=*Medium* and Balance=*Imbalanced*	Weight= *Medium*
Size=*Big* and Balance=*Balanced*	Weight= *Big*
Size=*Small* and Balance=*Balanced*	Weight= *Medium*
Size=*Big* and Balance=*Highly Imbalance*	Weight= *Medium*
Size=*Medium* and Balance=*Balanced*	Weight= *Big*

TABLE II. EXPERIMENTAL RESULTS. BOLD NUMBERS REPRESENTS BEST RESULTS FOR EACH CLIENT FOR EACH METRIC

Algorithm	Metric	Client 1	Client 2	Client 3	Client 4	Client 5	Average
FedSGD with original weights	Accuracy	0.70	0.77	0.69	0.68	0.81	0.73
	AUC Score	0.717	0.714	0.679	0.713	0.684	0.702
	F1 Score	0.71	0.77	0.70	0.69	0.80	0.734
FedSGD with FIS weights	Accuracy	0.71	0.77	0.71	0.70	0.79	0.736
	AUC Score	0.715	0.713	0.698	0.719	0.678	0.704
	F1 Score	0.71	0.76	0.71	0.71	0.77	0.732
FedND with original weights	Accuracy	0.74	0.81	0.78	0.72	**0.84**	0.778
	AUC Score	0.763	**0.776**	**0.781**	**0.762**	**0.733**	**0.763**
	F1 Score	0.75	**0.82**	0.78	0.73	**0.84**	**0.784**
FedND with FIS weights	Accuracy	**0.76**	**0.81**	**0.78**	**0.74**	0.82	**0.782**
	AUC Score	**0.763**	0.762	0.768	0.761	0.705	0.751
	F1 Score	**0.76**	0.81	**0.78**	**0.74**	0.81	0.78

V. CONCLSUION

In this work, tested the performance of FedSGD with FedND where we calculate direction in client and aggregate them in the server. We also compared the performance different weighting schemes. Obtained results showed that FedND outperformed FedSGD in every evaluation metric. FIS based weights shows slightly better results in accuracy score, but in other metrics results were not consistent. The results shows that second order method are better than first SGD based methods can be successfully applied in federated learning environment. Although FIS based weighting didn't show better results in FedND algorithm, it outperformed FedSGD with original weighting scheme.

However, there are still room for improvement of the suggested methods. One of the bottlenecks of FedND of these approaches was its complexity. Calculation of Hessian is costly and slows down the process. This difficulty can be tackled by Quasi-Newton methods by approximating the hessian. Another possible future work includes applying FedND as in the FedAVG where instead sending direction, we can update the model parameters several times using Newton method before sending it to server for aggregation

REFERENCES

[1] N. Saeed, H. Malik, A. Naeem, and U. Bashir, "Incorporating big data and IoT in intelligent ecosystems: state-of-the-arts, challenges and opportunities, and future directions," Multimedia Tools and Applications, Aug. 2023, doi: https://doi.org/10.1007/s11042-023-16328-3.

[2] Samar Samir Khalil, N. S. Tawfik, and M. Spruit, "Exploring the potential of federated learning in mental health research: a systematic literature review," Applied Intelligence, Jan. 2024, doi: https://doi.org/10.1007/s10489-023-05095-1.

[3] P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino, and F. Piccialli, "Model aggregation techniques in federated learning: A comprehensive survey," Future Generation Computer Systems, vol. 150, pp. 272–293, Jan. 2024, doi: https://doi.org/10.1016/j.future.2023.09.008.

[4] M. Y. Balik, "Comparing Federated Stochastic Gradient Descent and Federated Averaging for Predicting Hospital Length of Stay," arXiv.org, 2024. https://arxiv.org/abs/2407.12741 (accessed Jan. 31, 2025).

[5] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," Proceedings of the National Academy of Sciences, vol. 118, no. 17, p. e2024789118, Apr. 2021, doi: https://doi.org/10.1073/pnas.2024789118.

[6] X. Yuan and P. Li, "On Convergence of FedProx: Local Dissimilarity Invariant Bounds, Non-smoothness and Beyond," Advances in Neural Information Processing Systems, vol. 35, pp. 10752–10765, Dec. 2022, Accessed: Jan. 31, 2025. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/45ecdd6cf1f507d378a3442ed89e580b-Abstract-Conference.html

[7] S. Aliyev and N. Ismayilova, "FL2: Fuzzy Logic for Device Selection in Federated Learning," 2023 IEEE 17th International Conference on Application of Information and Communication Technologies (AICT), pp. 1–6, Oct. 2023, doi: https://doi.org/10.1109/aict59525.2023.10313160.

[8] S. Aliyev, "Application of AHP for Weighting Clients in Federated Learning," Azerbaijan Journal of High Performance Computing, vol. 6, no. 2, pp. 153–162, 2023, doi: https://doi.org/2616-6127%202617-4383.

[9] S. Bischoff, S. Günnemann, M. Jaggi, and S. U. Stich, "On Second-order Optimization Methods for Federated Learning," arXiv.org, 2021. https://arxiv.org/abs/2109.02388 (accessed Jan. 31, 2025).

[10] Anis Elgabli, Chaouki Ben Issaid, A. S. Bedi, Ketan Rajawat, M. Bennis, and V. Aggarwal, "FedNew: A Communication-Efficient and Privacy-Preserving Newton-Type Method for Federated Learning," PMLR, pp. 5861–5877, Jun. 2022, Accessed: Jan. 31, 2025. [Online]. Available: https://proceedings.mlr.press/v162/elgabli22a.html

[11] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection Against Reconstruction and Its Applications in Private Federated Learning," arXiv.org, Jun. 03, 2019. https://arxiv.org/abs/1812.00984

[12] K. Yang, T. Fan, T. Chen, Y. Shi, and Q. Yang, "A Quasi-Newton Method Based Vertical Federated Learning Framework for Logistic Regression," arXiv.org, 2019. https://arxiv.org/abs/1912.00513 (accessed Jan. 31, 2025).