МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ В ПРИРОДНИЧИХ НАУКАХ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

MATHEMATICAL MODELING IN NATURAL SCIENCES AND INFORMATION TECHNOLOGIES

DOI: 10.31319/2519-8106.1(50)2024.304775
UDC 519.6

Volosova Nataliia, Candidate of Technical Sciences, Associate Professor, Department of Mathematical Modeling and System Analysis
Волосова Н.М., кандидат технічних наук, доцент, кафедра математичного моделювання та системного аналізу
ORCID: 0000-0002-1314-1991
e-mail: nataliavolosova11@gmail.com

Tkachuk Mykyta, master's degree student, Department of Mathematical Modeling and System Analysis
Ткачук М.О., здобувач другого (магістерського) рівня, кафедра математичного моделювання та системного аналізу
e-mail: nikita.tkachuk59@gmail.com

Dnipro State Technical University, Kamianske
Дніпровськийдержавнийтехнічнийуніверситет, м. Кам’янське

MATHEMATICAL MODELING OF FEDERAL LEARNING BY SIMPLE ITERATION METHOD

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ФЕДЕРАТИВНОГО НАВЧАННЯ МЕТОДОМ ПРОСТОЇ ІТЕРАЦІЇ

In the study, modifications of well-known federated learning algorithms were developed and their experimental verification was performed, strategies for reducing the volume of communication between devices were proposed. The performed modifications are aimed at obtaining a better result of model training with a smaller number of communications between the device and the server. Thanks to the obtained model, the speed of convergence is increased while the communication rounds are reduced, and strategies for further optimization of federated learning algorithms are proposed.
***Keywords**: federated learning, fixed point, simple iteration, gradient method, optimization, non-stretching operator.*

У роботі описано математичну модель федеративного навчання та пошуку нерухомих точок методом простої ітерації та виконано оцінку ефективності модифікацій алгоритмів федеративного навчання, складених за даною моделлю.

При виконанні дослідження було проаналізовано існуючі підходи до технології федеративного навчання; досліджено відомі алгоритми федеративного навчання, визначено їх різно-

види та властивості, проведено експериментальну оцінку їх ефективності. Розроблено модифікації існуючих алгоритмів федеративного навчання, напрямлені на отримання кращого результату навчання моделі при меншій кількості комунікацій між пристроєм та сервером, що є одним з основних напрямків покращення наявних алгоритмів. Описано дві стратегії в розробці алгоритмів федеративного навчання для пошуку нерухомих точок: перша базується на фіксованій кількості локальних кроків, а друга базується на рандомізованих обчисленнях. В обох випадках метою стратегії є обмеження обміну локально обчисленими параметрами, що часто є вузьким місцем у розподілених структурах. Метою рандомізованих комунікацій є впровадження варіативності та стохастичності в процес комунікації та агрегації. Ця випадковість може служити кільком важливим цілям. Наприклад, рандомізовані комунікації можуть підвищити надійність алгоритмів федеративного навчання в сценаріях, коли дані між пристроями розподіляються неідентично. Завдяки введенню випадковості модель може адаптуватися до різноманітних розподілів даних, зменшуючи ризик перепідгонки до конкретних шаблонів, присутніх в окремих пристроях. Також рандомізовані комунікації можуть сприяти збереженню конфіденційності, запобігаючи зловмисникам розпізнавати конфіденційну інформацію про окремі пристрої на основі переданих оновлень.

Було виконано експериментальну перевірку розроблених модифікацій алгоритмів федеративного навчання тапідтверджено їх ефективність.

Завдяки отриманій математичній моделі збільшена швидкість збіжності при зменшенні раундів зв'язку, запропоновано стратегії для подальшої оптимізації алгоритмів федеративного навчання.

***Ключові слова**: федеративне навчання, нерухома точка, проста ітерація, градієнтний метод, оптимізація, нерозтягуючий оператор.*

Problem's formulation

Recently, the transition to distributed computing has become increasingly relevant, as modern hardware increasingly relies on the parallel integration of many separate units into a single system. The huge number of mobile phones or smart home devices in the world contain a significant amount of data received and stored on each of them. These data contain a wealth of potentially useful information for their owners, especially if appropriate machine learning models can be trained on the heterogeneous data stored in a network of such devices. However, many users are increasingly sensitive to privacy concerns and prefer that their data never leave their devices. But the only way to share knowledge without having all the data in one place is through communication. Typically, mobile phones exchange data with a remote server, whereby the global model gradually improves and approaches a steady state that is globally optimal for all users. This is precisely the goal of the federated learning paradigm. Communication, which can be expensive and slow, is a major bottleneck in this system. Therefore, it is extremely important to develop new algorithms where the computational and communication load is balanced. The problem of the impossibility of collecting all data on one device is quite urgent, therefore the issue of modeling federated learning is relevant.

Analysis of recent research and publications

In the work [3] Grigory Malinovsky, Dmitry Kovalev, Elnur Gasanov, Laurent Condat, Peter Richtárik "From local SGD to local fixed-point methods for federated learning" 2 main algorithms of federated machine learning are described. In the first algorithm, a clear sequence of synchronization moments is specified. In the second algorithm, synchronization occurs with a given probability.

The performed modifications are aimed at obtaining a better result of model training with a smaller number of communications between the device and the server. Now, this is one of the main directions for improving existing algorithms. After all, when we have a large number of devices, such communications significantly slow down the operation of the distributed system.

Formulation of the study purpose

The purpose of the research is the mathematical modeling of federated learning and the search for fixed points by the method of simple iteration and the evaluation of the effectiveness of modifications of federated learning algorithms.

To achieve the goal, the following tasks were formulated:

1) analyze existing approaches related to federated learning technology;
2) research federated learning algorithms, establish their varieties and properties;
3) evaluate the effectiveness of modifications to existing federated learning algorithms based on the simple iteration method.

Presenting main material

Federated learning aims to create a federated machine learning model based on data located on different devices [1]. There are two processes in federated learning: model training and model inference. In the process of training the model, the parties can exchange information, but not all data. The exchange does not expose any protected private parts of each device's data. A trained model can reside in a single device or be shared by multiple devices. In inference, the model is applied to new data. In a broad sense, federated learning is an algorithmic infrastructure (framework) for creating machine learning models, which can be characterized as follows: a model is a function whose argument is a data instance (Sample) from some device, and the value of the function is the result (Outcome).

More formally, consider *N* data stores $\{F_i\}_{i=1}^N$, that wish to train an *ML* model using their respective datasets $\{D_i\}_{i=1}^N$. The traditional approach is to collect all data $\{D_i\}_{i=1}^N$ together on the same data server and train the model M_{SUM} on the server using a centralized dataset. In the traditional approach, any data owner F_i will provide their data to the server D_i in such a way that the server and even other users will have access to it.

Federated learning is a *ML* process, in which data owners jointly train a model M_{FED} without collecting all the data $\{D_i\}_{i=1}^N$. We will also denote v_{SUM} and v_{FED} the performance indicators (for example, accuracy, completeness, F_1-measure) of the centralized model M_{SUM} and the combined model M_{FED}, respectively.

Let δ is a non-negative real number, then the federated learning model M_{FED} has δ — loss of efficiency if $|v_{SUM} - v_{FED}| < \delta$.

This inequality expresses the following intuition: if federated learning is used to build a*ML* model on distributed data stores, the model's performance on future data will be approximately the same as a model built on pooling all data.A federated learning system may or may not include a central coordinating computer depending on the need. With federated learning, there is no need to create a central database, and any financial institution that participates in federated learning can initiate new user requests to other institutions within that federation. Other agencies only need to obtain information about a person's credit without knowing specific information about the person. This not only protects user privacy and data integration, but also detects multilateral lending.

Classic machine learning is basically characterized by one iterative process $x^{k+1} = x^k$.

There are many such iterative processes in federated training. Each of them takes place on its own device / node / node with its own local data. Fig. 1 shows this architecture.

In the figure F_i is a local operator that specifies a simple iteration on the device*i*. If the number of devices is equal to *M*, then the iterative process of each of them can be described as follows:

$$x_i^{k+1} = F_i x_i^k,$$
$$x_i^{k+1} = x_i^k + \gamma_i \nabla f_i(x_i^k),\ i = 1, 2, ..., M.$$

It is clear from the description of the procedure that each device has its own current solution x_i^k, its own function f_i, its own step γ_i.

In federated learning, the main goal is to obtain an "integral" solution — a solution that would be suitable for the entire set of data, and not for each individual node. For this, with some frequency, all nodes exchange their solutions with the master node. The master node averages them, and then broadcasts the averaged solution to all nodes. That is, after averaging, all nodes receive one current solution. After receiving, each node goes its own way, stepping at its own pace γ_i, in the direction of its gradient ∇f_i. This happens until the moment of re-averaging. Important is the fact that only solutions are exchanged, no data is exchanged. Data is stored locally.

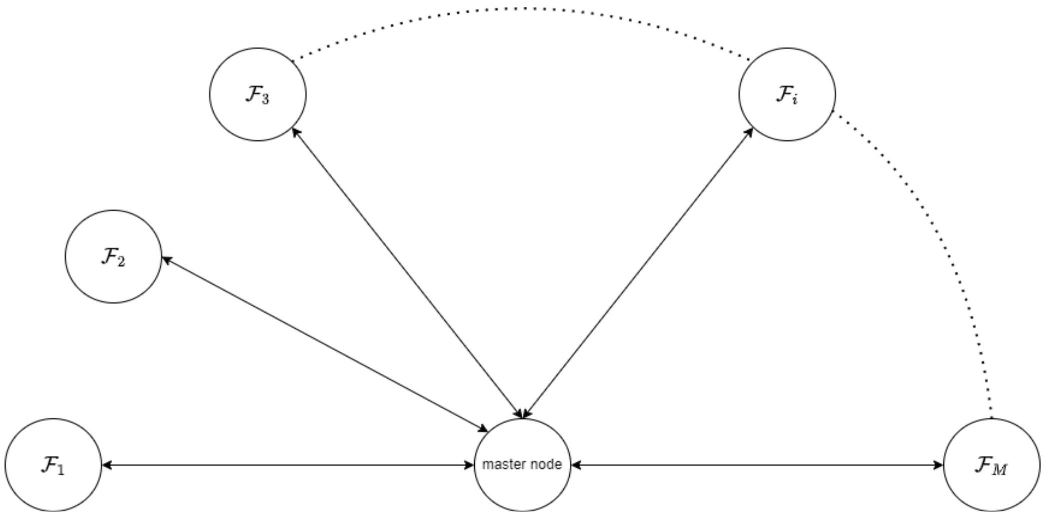

Fig. 1. The classical architecture of federated learning

The communication frequency of all nodes with the master node can be set differently. One of the simplest options is to create a sequence of moments t_n, in each of which the nodes will contact the master. In real life, the sequence can be hourly. For example, the *i*-th device sends its current resolution every hour. After sending, with some delay, he receives an averaged solution, from which he continues his calculation. Another option of communication is that the server has its own counter, which at a certain moment in time, with probability*P*, sends a signal to all devices to collect their current solutions. In this study, algorithms were developed for both communication options.

Let us consider in detail the option when the averaging occurs in a certain sequence of moments t_n. This sequence will be determined by the periodicity *H>0:* $t_n = nH, n \in N$.

We have a common iteration counter $k = 0, 1, 2, \ldots$. At the moment when $k \in t_n$ the averaging will take place

$$\bar{x}^k = \frac{1}{M}\sum_{i=1}^{M} x_i^k .$$

After averaging, the next node iteration *i*will look like this: $x_i^{k+1} = \bar{x}^k + \gamma_i \nabla f_i(\bar{x}^k)$.For further analysis, let's write down the operator used at each *H*steps:

$$\bar{F} = \frac{1}{M}\sum_{i=1}^{M}(F_i)^H ,$$

here F_i is the operator responsible for the next iteration of each node; *H* is the composition of the operator with itself.

During the period *H*, the moments of connection will look like $t_n = \{H, 2H, 3H, \ldots, nH\}$. Using the operator $\bar{F}$, the iterative process of all nodes can be presented in the form: $\bar{x}^{(n+1)H} = \bar{F}(\bar{x}^{nH})$.

The given mathematical model is used for modifications of the algorithms developed in the study.

The algorithms proposed in [3] work as follows: at each iteration, an operator T_i is applied to the node *i*with the parameter λ. After a certain number of iterations, each of the computing *M* nodes transmits its vector to the master node, which calculates their average value and broadcasts it to all nodes. Thus, a separate node continues the calculation on the next iteration from the same variable x^k. The algorithms are a generalization of local gradient descent, also known as FedAvg.

In the conducted study, modifications of algorithms 1, 2 were developed in order to reduce the number of nodes communicating with the master, that is, to reduce the number of communications in the federated learning system.

The created algorithm 3 is a modification of algorithm 1, which implements cyclic communication of a random subset of devices with the master node.

The modification is as follows:

1) An additional parameter s is specified, which determines the percentage of all devices communicating with the master.

2) At each averaging, r nodes are randomly selected and then these r nodes are connected to the master.

It is expected that thanks to this algorithm, it is possible to significantly save on the number of connections, without significantly losing the speed of convergence.

Algorithm 3 is shown in Fig. 2.

Algorithm 3 Циклічний зв'язок рандомної підмножини пристроїв з master node

Input: stepsize $\lambda > 0$, communication times t_n, percent of nodes for communication s
Initialize: start point w_i^0 for each node
for $k = 1, 2, ..$ **do**
  for $i = 1, 2, ..., M$ **do**
    if $k - 1 \in t_n$ **then**
      Translate previous step averaging to nodes
      $w_i^{k-1} = \overline{w}^{k-1}$
    end if
    $w_i^k = (1-\lambda) w_i^{k-1} + \lambda \mathcal{F}_i(w_i^{k-1})$
  end for
  if $k \in t_n$ **then**
    Select random $r = \frac{s}{100} \cdot M$ nodes
    We connect all r nodes with master
    Average selected r nodes values
    $\overline{w}^k = \frac{1}{M} \sum_{i=1}^{M} w_i^k$
  end if
end for

Fig. 2. Algorithm 3

The developed algorithm 4 is a modification of algorithm 2 with the specified transformations and implements the random connection of a random subset of devices with the master node. Algorithm 4 in pseudocode is shown in Fig. 3.

The resulting modifications aim to reduce the volume of transmitted information, which in turn will help increase the speed of communication between the server and the device.

The computational experiment conducted in the study consists of testing algorithms on the same dataset. The goal is that with fewer connections to the master node, the algorithm converges faster. A metric is taken as a convergence estimate $\left\| \overline{x}^{(n+1)H} - \overline{x}^{nH} \right\|$.

In the course of the experiment, logistic regression was used, which is one of the most popular models for solving classification problems, in this case, the problem of binary classification [11].

Each device has its own error function $f(x)$:

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + \exp\left(-y_i w^T x_i\right)\right) + \lambda \|x\|^2 .$$

Algorithm 4 Випадковий зв'язок рандомної підмножини пристроїв з master node

Input: stepsize $\lambda > 0$, communication probability p
Initialize: start point w_i^0 for each node
for $k = 1, 2, ..$ **do**
  for $i = 1, 2, ..., M$ **do**
    if $t_{k-1} > 1 - p$ **then**
      Translate averaging to nodes
      $w_i^{k-1} = \overline{w}^{k-1}$
    end if
    $w_i^k = (1 - \lambda) w_i^{k-1} + \lambda \mathcal{F}_i(w_i^{k-1})$
  end for
  t_k = generate random $t \in [0, 1]$
  if $t_k > 1 - p$ **then**
    Select random $r = \frac{s}{100} \cdot M$ nodes
    We connect all r nodes with master
    Average selected r nodes values
    $\overline{w}^k = \frac{1}{M} \sum_{i=1}^{M} w_i^k$
  end if
end for

Fig. 3. Algorithm 4

By structure, this function is the same for all devices. The local dataset consists of observationsx_i. The answers to the observations are $y_i \in \{-1, +1\}$. The coefficients inside the regression are recorded in the vector w. An iterative process takes place on each device:

$$x^{k+1} = x^k - \gamma \nabla f(x^k),$$

here$\gamma$is the step size.

For software implementation, the package«sklearn» [14] was used. Number of nodes are $M=10$.The data was taken from the a9a dataset [15], which has 32,000 observations with 123 parameters and is intended for testing binary classification methods.

The effectiveness of algorithms 1 and 2 was studied in [3]. In the conducted research, these algorithms were implemented and run on the a9a dataset, and results consistent with the results of [3] were obtained.

The cyclic communication with the master node, implemented in algorithm 1, is more stable, and in the second algorithm, the convergence is more volatile than in the first one due to the randomness of the communication with the master node. The results of the study on the convergence of algorithms 1 and 2 are shown in Fig. 4 and 5, respectively.

When reproducing algorithms 1 and 2, the accuracy is already 0.0001 at the thousandth iteration.

These modifications simulate cases in which not all nodes are connected, but only a certain percentage of them (a subset). Moreover, the subset is chosen randomly, with return. Thus, two identical nodes can be present in this set. According to the graph, algorithm 3 of cyclic communication appears to be stable, with good accuracy, starting from the thousandth iteration we have a difference of less than 0.005. Algorithm 4 with a random link, on the contrary, is volatile. In addition, the accuracy is much lower, on average the metric is 0.4. The results of checking the convergence of the algorithms are shown in fig. 6 and 7. In the presented graph of algorithm 3, 80 % of all nodes exchanged with the server. At this percentage, the accuracy almost did not deteriorate, at the thousandth iteration the accuracy is 0.005. At the same time, the number of devices sending data to the server decreased by 20 % [16, 17].

Thus, two strategies can be distinguished in the development of federated learning algorithms for finding fixed points: the first is based on a fixed number of local steps, and the second is based on randomized calculations. In both cases, the goal of the strategy is to limit the sharing of locally computed parameters, which is often a bottleneck in distributed frameworks. The purpose of randomized communications is to introduce variability and stochasticity into the process of communication and

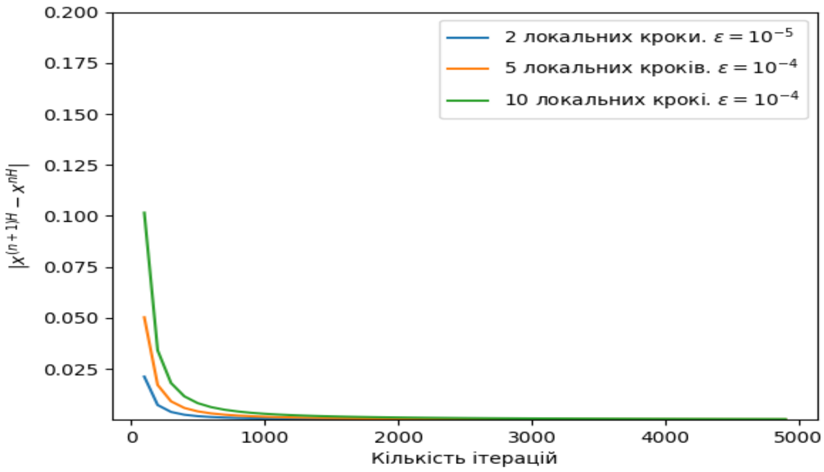

Fig. 4. Convergence of algorithm 1

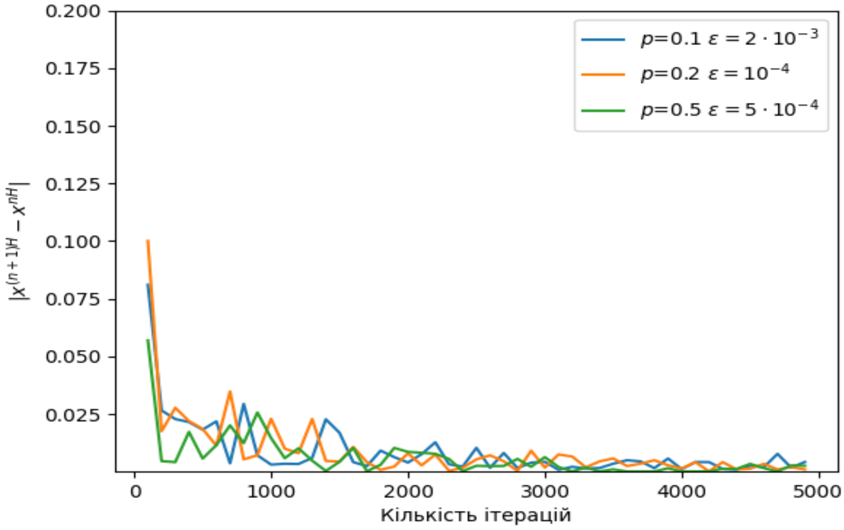

Fig. 5. Convergence of algorithm 2

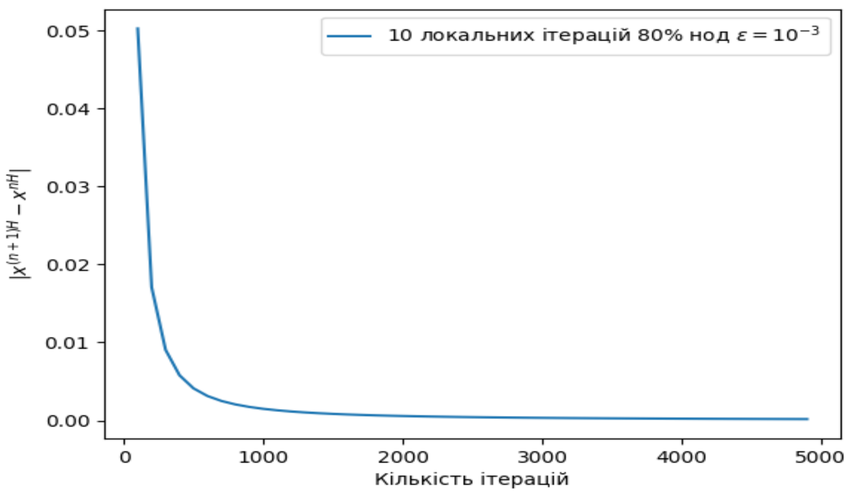

Fig. 6. Convergence of algorithm 3

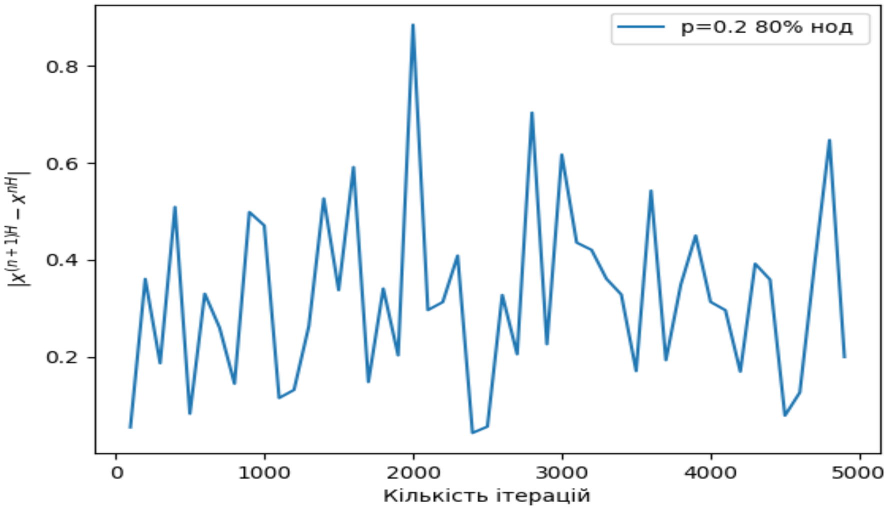

Fig. 7. Convergence of algorithm 4

aggregation. This randomness can serve several important purposes. For example, randomized communications can improve the robustness of federated learning algorithms in scenarios where data is distributed non-identically (non-IID) between devices. By introducing randomness, the model can adapt to diverse data distributions, reducing the risk of overfitting to specific patterns present in individual devices. Randomized communications can also help preserve privacy by preventing attackers from identifying sensitive information about individual devices based on transmitted updates. Randomness hides the connection between specific updates and the devices that created them, adding a layer of privacy protection.

Conclusions

The work developed and conducted an experiment in which Algorithm 1 of cyclic communication with the master node and Algorithm 2 of random communication with the master node were reproduced, the effectiveness of algorithms 1 and 2 was confirmed. When reproducing these algorithms, the accuracy already at the thousandth iteration is 0.0001.

Modifications of algorithms 1, 2: algorithms 3 and 4 were created, the purpose of which was to reduce the number of rounds of communication between the master node and devices, without significantly losing accuracy.

In further research, it is necessary to additionally analyze the developed algorithms. The question of how the behavior of algorithms will change with a rapid increase in the number of nodes requires research. How will the behavior of the algorithms change if you dial nodes without returning? How will algorithms behave on different datasets?

The proposed approaches can be applied to further improve federated learning algorithms, to study the effectiveness of algorithms, both with cyclical and random communication, and to design experiments to confirm the effectiveness of algorithm modifications.

References

[1] Federated learning: collaborative machine learning without centralized training data. Google Research Blog. URL: https://blog.research.google/2017/04/federated-learning-collaborative.html

[2] Contributors to Wikimedia projects. Federated learning - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Federated_learning

[3] Grigory Malinovsky, Dmitry Kovalev, Elnur Gasanov, Laurent Condat, Peter Richtárik (2020) From local SGD to local fixed-point methods for federated learning.Cornell University//arXiv. URL: https://arxiv.org/abs/2004.01442v2.

[4] Nick Goudl Linesearch methods for unconstrained optimization. MSc course on nonlinear optimization URL: https://www.numerical.rl.ac.uk/people/nimg/msc/lectures/
[5] Banach fixed-point theorem. URL: https://en.wikipedia.org/wiki/Banach_fixed-point_theorem
[6] Ryu E. K., Yin W. Large-Scale (2022) Convex Optimization: Algorithm Analysis Via Monotone Operators. Cambridge University Press, 319 p.
[7] Bauschke H. H., Combettes P. L. (2017) Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer,619 c.
[8] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang (2016) Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS ’16).
[9] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong (2018) Federated Learning. Communications of The CCF 14, 11, pp 49–55.
[10] Raad Bahmani, Manuel Barbosa, Ferdinand Brasser, Bernardo Portela, Ahmad-Reza Sadeghi, Guillaume Scerri, and Bogdan Warinschi (2017) Secure Multiparty Computation from SGX. In Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, Revised Selected Papers,April 3-7, 2017, pp 477–497.
[11] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti (2018) A Survey on Homomorphic Encryption Schemes: Theory and Implementation. ACM Comput. Surv. Vol. 51, Sssue 4, Article 79, pp 1–35.
[12] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra (2018) Federated Learning with Non-IID Data. arXiv:1806.00582[cs.LG]
[13] Sinno Jialin Pan and Qiang Yang (2010) A Survey on Transfer Learning. IEEETransactions on Knowledge and Data Engineering, Vol. 22, No. 10, pp 1345–1359.
[14] sklearn.linear_model.SGDClassifier https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.GDClassifier.html
[15] LIBSVM Data: Classification (Binary Class) https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#a9a
[16] TkachukN., Volosova N. (2023) Federal learning algorithms based on searching for fixed points by simple iteration method// Problems of mathematical modeling: materials of the All-Ukrainian science and method. conference, May 24-26, 2023. Kamianske: DSTU, pp 93-94.
[17] Volosova N., Tkachuk N. (2023) Mathematical modeling of federal learning by simple iteration method//Abstracts of XII International Scientific and Practical Conference. "Youth, education and science through today’s challenges", December 04-06, 2023, Bordeaux, France,pp 360-362.URL: https://eu-conf.com/ua/events/youth-education-and-science-through-today-s-challenges/

Список використаної літератури

1. Federated learning: collaborative machine learning without centralized training data. Google Research Blog. URL: https://blog.research.google/2017/04/federated-learning-collaborative.html (дата звернення: 06.10.2023).
2. Contributors to Wikimedia projects. Federated learning - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Federated_learning (дата звернення: 05.10.2023).
3. Grigory Malinovsky, Dmitry Kovalev, Elnur Gasanov, Laurent Condat, Peter RichtárikFrom local SGD to local fixed-point methods for federated learning.Cornell University//arXiv. 2020. URL: https://arxiv.org/abs/2004.01442v2(дата звернення: 01.12.2023).
4. Nick Goudl Linesearch methods for unconstrained optimization. MSc course on nonlinear optimization URL: https://www.numerical.rl.ac.uk/people/nimg/msc/lectures/(дата звернення: 1.12.2023).
5. Banach fixed-point theorem. URL: https://en.wikipedia.org/wiki/Banach_fixed-point_theorem (дата звернення: 07.10.2023).

6. Ryu E. K., Yin W. Large-Scale Convex Optimization: Algorithm Analysis Via Monotone Operators. Cambridge: Cambridge University Press, 2022. 319 p.
7. Bauschke H. H., Combettes P. L. Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, 2017. 619 p.
8. Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. October 2016. P. 308–318.https://doi.org/10.1145/2976749.2978318
9. Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated Learning. Communications of The CCF 14, 11. 2018. P.49–55.
10. Raad Bahmani, Manuel Barbosa, Ferdinand Brasser, Bernardo Portela, Ahmad-Reza Sadeghi, Guillaume Scerri, and Bogdan Warinschi. Secure Multiparty Computation from SGX. In Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, Revised Selected Papers. April 3-7, 2017. P.477–497.
11. Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. ACM Computing Surveys.Volume 51. Issue 4. Article No.79. (July 2018). P.1–35.https://doi.org/10.1145/3214303
12. Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated Learning with Non-IID Data. 2018. arXiv:1806.00582 [cs.LG]https://doi.org/10.48550/arXiv.1806.00582(дата звернення: 2.12.2023).
13. Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, Vol. 22, No. 10, October 2010. P.1345–1359.
14. sklearn.linear_model.SGDClassifier https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html (дата звернення: 4.12.2023).
15. LIBSVM Data: Classification (Binary Class) https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#a9a (дата звернення: 4.12.2023).
16. Ткачук Н., Волосова Н. Federal learning algorithms based on searching for fixed points by simple iteration method// Проблеми математичного моделювання: матеріали Всеукр. наук.-метод. конф., 24-26 трав. 2023 р. Кам'янське: ДДТУ, 2023. С. 93-94.
17. Volosova N., Tkachuk N. Mathematical modeling of federal learning by simple iteration method//Abstracts of XII International Scientific and Practical Conference. "Youth, education and science through today's challenges", December 04-06, 2023, Bordeaux, France. 2023. P. 360-362.URL: https://eu-conf.com/ua/events/youth-education-and-science-through-today-s-challenges/

Надійшла до редколегії 10.01.2024