Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/22/20231215

Problems, solutions and improvements on federated learning model

Leqi Huang

Big Bridge Academy, Wuxi, Jiangsu, China, 214000,

lambert_huang1123@163.com

Abstract. The field of machine learning has been stepping forward at a significant pace since the 21century due to the continuous modifications and improvements on the major underlying algorithms, particularly the model named federated learning (FL). This paper will specifically focus on the Partially Distributed and Coordinated Model, one of the major models subject to federated learning, to provide an analysis of the model's working algorithms, existing problems and solutions, and improvements on the original model. The identification of the merits and drawbacks of each solution will be founded on document analysis, data analysis and contrastive analysis. The research concluded that both alternative solutions and improvements to the original model can possess their unique advantage as well as newly-emerged concerns or challenges.

Keywords: machine learning, federated learning, partially distributed and coordinated model, local epoch adjustment.

1. Introduction

The terminology "Partially Distributed and Coordinated Model" illustrated the characteristics of the model--some of the tasks are distributed across multiple devices, while others are coordinated centrally. Current researches and solutions on this particular model's problems focus mainly on reducing the time required for accomplishing model updates, leaving relatively little attention on the drawbacks brought by the improvement of updating speed. This paper therefore will identify several improvements to the current solutions to give consideration to both time and other concomitant issues. This paper can contribute to further research and applications in the field of machine learning by clarifying the benefits and disadvantages of each solution and improvement, making the selection process explicit and convenient for designers while adopting models to solve problems under real-world circumstances.

2. Model foundations and problems

The partially distributed and coordinated model consists of a centralized cloud server and multiple client devices. The server will collect data, process information and distribute models to be passed on to clients, who will execute model training using a local database. As shown in figure 1, When being compared with a centralized and coordinated model, which refers to the scenario that local devices will send the raw data, instead of models, to the cloud server for processing, partially distributed model illustrated its significant improvement on privacy concerns since the server cannot have direct access to raw data, which are processed locally.

DOI: 10.54254/2755-2721/22/20231215

Figure1. An example of federated model's application on predicting the next typing word on mobile phones [1].

However, the problem regarding the time required to complete updates emerged while the privacy issue is being solved using this model. Since the general update executed in the server will not initiate until all the local updates have been uploaded, the whole system's working speed is depending on the client device that is updating the slowest. Meanwhile, the heterogeneity of devices makes the difference in processing speed almost unavoidable: The devices' capabilities of storing, computing and communicating may vary significantly due to the variability in hardware, like CPU, network conditions and battery level. For instance, when device failure, a situation when a device fails to accomplish updates due to Internet malfunctioning or other reasons, occurs, the total training time required to converge to a global model will be extended. Meanwhile, there are devices underrepresented by the global model because they seldom participate in the convergence process thanks to their abnormal states like CPU busy [2].

3. Existing solutions

3.1. Local multiple updates

Given the problems of the crude decentralized and coordinated model, two major solutions have been provided to shorten the time required for each communication round. The first solution provided a novel insight into the problem by performing multiple local updates until the overall time for one update is approximately equal for all clients. In this scenario, while the interval for updating may even be larger than the original model, the faster devices are able to perform more updates locally instead of merely waiting for the slowest one to finish a single update. Multiple local updates, when appropriately adopted, can further enhance the overall performance because of the more accurate models updated to the server. However, it is also worth mentioning that at the time when local updates are performed at unreasonable times, the problem of overfitting can emerge to greatly compromise the resulting model concluded by the server. Overfitting refers to the scenario that the model is overly conformed to the training dataset and therefore cannot be generalized. And one of the major reasons for the occurrences of overfitting is that the model is trained on a single sample set of data for an overly long time, resulting in the model's absorption of irrelevant information in the database [3].

DOI: 10.54254/2755-2721/22/20231215

3.2. Asynchronous simple gradient descent

Another proposed solution is named Asynchronous Simple Gradient Descent (Asynchronous SGD), which, as illustrated by the name, suggests individual clients update the server once their local processing course has been concluded. Adopting this strategy enables the devices with higher computational speed to proceed training without having to wait for the slower ones to complete a global update. However, another problem identified as update imbalance can be caused by updating models independently: Since the faster devices are continuously updating models at a speed higher than the slower ones, the models that cloud server will receive are primarily occupied by models generated by faster clients. Therefore, the clients with higher computing speed can unconsciously weigh more in the ultimate general model than those with medium or moderate capabilities, resulting in worse optimization performance in a system with relatively imbalanced data distribution [4].

4. Improvements to the original algorithm

4.1. Variable workload

As the commonly-adopted solutions mentioned above are possessing both advantages and drawbacks, improvements on the original algorithm can also be made aiming to achieve efficiency as well as perfection. The first strategy employs deep reinforcement learning (DRL) network to derive the optimal policy for local workload adjustments. The only difference is that variable workload allows the server to distribute data to devices according to their computational power, with faster devices receiving more data and slower ones bearing less burden. The updates can then be conducted synchronously and therefore avoid time latency created by the heterogeneity of devices. Also, adjusting the local workload ruled out the possibility of overfitting when doing local updates and being biased when adopting the faster devices' models continuously. But the challenge of accurately predicting the models' training time can also arise due to the variation in the number of epochs on each device as well as the heterogeneity in the structure of the models. A method to predict the training time in accuracy is therefore proposed by Justus et al., whose fundamental idea for the algorithm is to utilize individual layers as the basic units for computation of the whole models' training time. By accumulating the training time, the variation in epoch quantities and model heterogeneity can be overlooked to a great extent [5].

4.2. Proximal term

Another modification to the original partially distributed and coordinated model is the addition of proximal terms on the local model training procedure. Scaled by μ, proximal terms are utilized to mitigate the effects of statistical and system heterogeneity by penalizing divergence from the global model. As a result, a large μ may hold the possibility to suppress the convergence process by overly penalizing divergence but a small μ may not be of significant effects. Therefore, an appropriate μ must be chosen for the specific approximation in order to guarantee convergence given the realistic environment in which heterogeneities could cause divergence. However, there remains the possibility for local updating schemes to have a worse performance than distributed simple gradient descent when the generated data are not identically distributed. Also, the problem regarding state changes of the devices remains unsolved with the addition of proximal terms. As a result, adding proximal terms to the original algorithm will not be enough to hold absolute superiority over other algorithms [6].

5. Conclusion

In conclusion, this paper focuses on identifying the merits and drawbacks of different algorithms applied in a partially distributed and coordinated model and it concluded that the alternative solutions, as well as improvements on the original algorithm, have respectively their advantages and drawbacks and therefore should be utilized after consideration and selection. The way of representing the characteristics can be improved by listing more experimental data for comparison. Future research in this area can be directed to better resolving the privacy concern by building a communication-efficient system despite the heterogeneity of devices.

DOI: 10.54254/2755-2721/22/20231215

References

[1] Tian L., Talwalkar T. (2019). Federated Learning: Challenges, Methods, and Future Directions. arXiv:1908.07873v1 [cs.LG] 21 Aug 2019

[2] Yang, C., Wang, Q., Xu, M., Chen, Z., Bian, K., Liu, Y., & Liu, X. (2020). Characterizing Impacts of Heterogeneity in Federated Learning upon Large-Scale Smartphone Data. arXiv preprint arXiv:2006.06983.

[3] Deshpande, S. (2021, October 5). Overfitting in ML: Avoiding the pitfalls. Towards Data Science. https://towardsdatascience.com/overfitting-in-ml-avoiding-the-pitfalls-d5225b7118d

[4] Diwangkara, S. S., & Kistijantoro, A. I. (2020). Study of data imbalance and asynchronous aggregation algorithm on federated learning system. https://ieeexplore.ieee.org/abstract/document/9264958/authors#authors

[5] Yan Zeng, Xin Wang, Junfeng Yuan, Jilin Zhang, Jian Wan, "Local Epochs Inefficiency Caused by Device Heterogeneity in Federated Learning", Wireless Communications and Mobile Computing, vol. 2022, Article ID 6887040, 15 pages, 2022. https://doi.org/10.1155/2022/6887040

[6] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks (Version 5) [Preprint]. arXiv. https://arxiv.org/abs/1812.06127