Modeling Asynchronous Federated Learning for Health care application

Vivek Khetade
Department of Electronic Design Technology, Shri Ramdeobaba College of Engineering & Management Nagpur , India

Abstract

Healthcare data from clinical institution and pharmaceutical industries are fragmented and are sensitive with protected health information of individual. Data privacy, safety and security are of prime important. It is vital to train machine learning model without compromising privacy. In federated learning the global model is aggregated on aggregation servers as per the parameter of local model instead of local data. There is waiting for straggler client before aggregation in each round .Every client operated at its own speed of training. Asynchronous federated learning aggregates the global model based on available updated model received at the server asynchronously during each round of training. Asynchronous federated learning improves the efficiency, performance, privacy and security. In this paper, based on the Lung cancer dataset, the classification model is built. With NVIDIA flare, synchronous federated learning is carried out with one server and three clients for spliited data of lung cancer of Kaggle. The performance of synchronous federated learning is observed similar to the classical machine learning model. Modeling of asynchronous federated learning is carried out with Petri Net and analyzed for the reachability analysis with workcraft software. Substantial improvement in medical data privacy, security and safety is observed.

Keywords : *Federated Learning , Asynchronous federated learning, Machine Learning, Artificial Intelligence*

DOI Number: 10.14704/nq.2022.20.9.NQ44248 **Neuro Quantology 2022; 20(9):2132-2138**

1 Introduction

Healthcare data comes from Hospital record, disease register, medical claims record, vital records such as birth, death, fetal death, cause of death, etc and medicinal supply chain network .Because of digitalization , this data is readily available .Quality of Healthcare delivery can be improved if we dig the data driven system. It provides opportunities for the development data driven computational system and technique. This data is fragmented and isolated. The reason of isolation is complex system of health care. Also there exists the local data protection regulation such as GDPR, HIPAA. It regulates the process of accessing and analyzing data. Personal health information of patient is covered by HIPAA privacy rule. Intelligent computing perceived advantage of using AI for Health care .Health care would be easier and quicker for more people to access. It also offers faster and more accurate diagnosis and makes better treatment. It makes it possible to have your own health care specialist anytime and on any device. Prominent and promising everyday clinical practice and medical research is offered by AI. AI requires large amount of high quality data for model trainer.

Hospitals, healthcare research and service provider required to work together. They must share and combine their knowledge by preserving their data privacy , security and safety. Isolated Data Island is created because of privacy concern. Multiple parties such as researcher, diagnostic center, imaging center, hospital can participate on ML model training without requiring direct access to their local training data. This is beauty of Federated learning (FL) [1][2][3]. The design goal of FL is to provide data privacy and develop satisfying ML models across multiple participants or stakeholders under the premise of legal compliance. When we compared the federated learning with traditional ML approaches, it offers various advantages (1) It provides higher level local data privacy due to weight aggregation of the global model (2) As the training data does

not need to be transferred to cloud it offers lower latency in network transmission. (3 due to the help of learned features from other client it provides high quality model. Paper is organized as follows, In section II system level behavior of synchronous federated learning is expressed along with its server side workflow and client side workflow. In section III, asynchronous federated learning is presented along with its algorithmic representation. Asynchronous behavior of AFL is expressed with Petri Net. Petri Net model is expressed for client side behavior and server side behavior. In section IV,details of the tools used for modeling of FL such as NVIDIA Flare and its component is expressed. Workcraft tool is also presented. In section V Result section is expressed where Petri Net model of AFL is analyzed for rechability analysis along with application of federated learning for medical application.

2 Synchronous federated Learning.

Breadan Mc Mahan atl introduced federated Learning [4].Final model is generated on an aggregation server by averaging the local models which are trained on distributed client. In each training round (iteration), client taking part in the training share their local models. Client does not share local training data. FL is considered as a ML framework. It removes the barrier of data silo because of privacy preserving features for local training data on each client.[5]The major steps of the synchronous FL are as follows.

(1) Initialization: Once the training job is determined to the medical application, the aggregation server prepares an initial model W_G^0 and training settings, such as the learning rate, batch size, and the number of iterations to train. After this selection, the aggregation server throw W_G^0 to an appropriate number of nodes (denoted as K).

(2) Local Model Training: Let **t** stand for the current iteration number. Based on final model W_G^t, each client trains its local model and obtains the local models W_k^t where k $k \in$ [1,k]. The local models are subsequently sent back to the aggregation server.

(3) Final Model Aggregation: The server aggregates local models and generates a new final model by:

$$W_G^{t+1} = \frac{1}{K}\sum_{k=1}^{K} W_k^{t\,k}$$

.After that, W_G^{t+1} is sent back to the client for the next iteration of training.

2.1 Synchronous Federated learning system Architecture

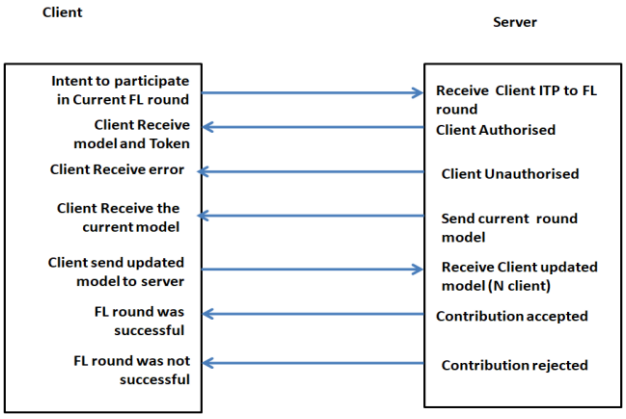

Fig 1.System Level architecture of Synchronous FL .

With reference to Fig .1 Federated Learning model training is started in two modes , secure mode and POC mode (proof of concept) .Generally secure mode is used for the deployment of model whereas POC mode can be used for the simulation of the concept federated learning. In an POC mode server is poll for the client request. Client start server training request through intent to participate in current FL round. Servers acknowledge the client credibility and authorized client. Server sends the initialized model to the client along with token. If client is not authorized, server will send the error message. Once the client received the model from the server, it start training with local data . Number of epochs to run is under the control of client, during each round of federated learning. Once number of epoch s is achieved of training on local data, client send the updated model to the server and poll for the message FL round was successful. Contribution is accepted by the server. Server updates model from all participating client by performing model aggregation for the current round. This updated model is again send to client for next round. This process continues till maximum round set on the

server are completed. This process is derived from NVIDIA Flare.[6]

2.2 Behavior of Server side of federated Learning

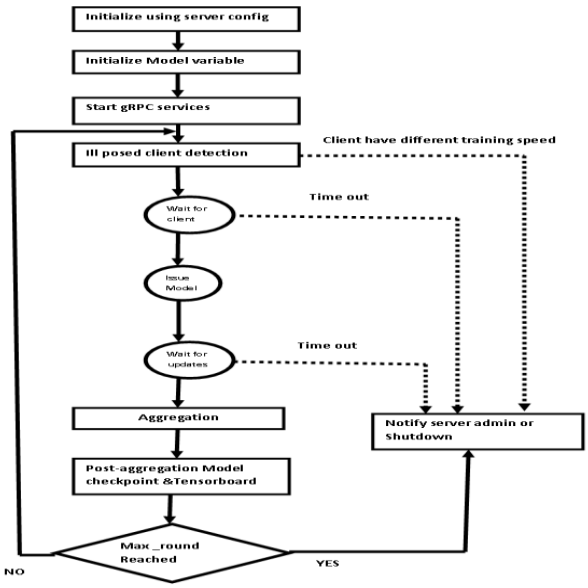

Fig 2. Behavior of aggregation Server of synchronous Federated Learning

As shown in Fig 2, server side services begins with initialization of initial model or restoring the model and configuration of server for maximum & minimum number of client along with communication port initialization and SSL certificate key. The server begins polling of client joining request. Once the request is received, server issues the model to the client and waits for client to send back the updated model. When server received the the required number of updated model ,it perform the aggregation based on weighted aggregation algorithm. It updates the current model and used for next round of model training.[6]

2.3 Behaviors of Client of Federated Learning

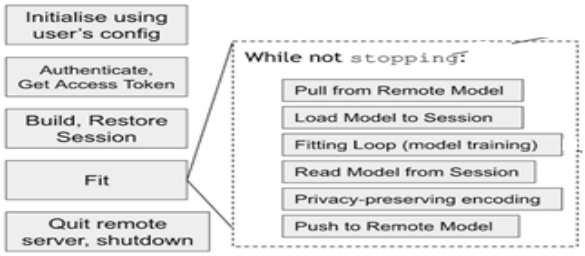

Fig 3 Behaviors of Client of Feerated Learning

As shown in Fig 3 in the beginning client configure to initialize the model. To make the login request to server and to get federated learning training token it uses credentials. When token is received, client request the current model. and built& restore the session. It starts the training on local training data. When completes training, it send the updated model to the server. Client can have his own privacy preserving policy. [6] It can have control over the weights to send back to the server. After this client can call for new global model to start new round of federated learning training.

3 Asynchronous federated Learning

To mitigate the influence of stale nodes and improve the efficiency of FL, asynchronous federated Learning model are designed. [5]The major steps of AFL are as follows.

(1) Initialization: Similar to synchronous FL, the aggregation server transmit the initial global model W_G^0

to all k nodes.

(2) Local Model Training: Nodes train their local models based on the most recent global model obtained from the aggregation server. Nodes finish training local models $\{W_1^t$, W_2^{t+1} , W_3^{t+2},..... $W_k^{t+k}\}$ in sequence due to the disparity of heterogeneous devices in computing capability. Then, the local models are independently sent back to the aggregation server.

(3) Final Model Aggregation: The server aggregates the newly collected local model with the latest final model by:

$$w_G^{t+k} = \frac{w_G^{t+k-1} + w_k^{t+k}}{2}.$$

After that, W_G^{t+k} is obtainable for nodes for the subsequent training process.

The current iteration number in AFL increments by 1 when a device finishes its training and uploads the

updated local model to the aggregation server. The immediate model aggregation in AFL reduces the waiting time and thereby improves efficiency[6][7][8][9][10].

3.1 Petri Net model of System Architecture of Asynchronous Federated Learning

The behavior of asynchronous federated learning is expressed with Petri Net (PN) diagram[11]. Client first start the server training service by giving the signal intent to participate (ITP) to the server and server send the received client intention (RCI) to the client. During this time the server authorized client and send the signal client authorized (CLA) .Client will send the signal client received ready(CRR).Server initialized and restore the model to be broadcast and then it will broad cast the model .client will receive the model and token for updating . After receiving model, client configure to initialize the model and build the model and restore the session and start the training of the model. When the required number of epoch has achieved, the client will send the request for sending updated model (RSUM).Client updated model is ready and it is indicated by sending updated model .server will acknowledge the received model by sending signal server received updated model.

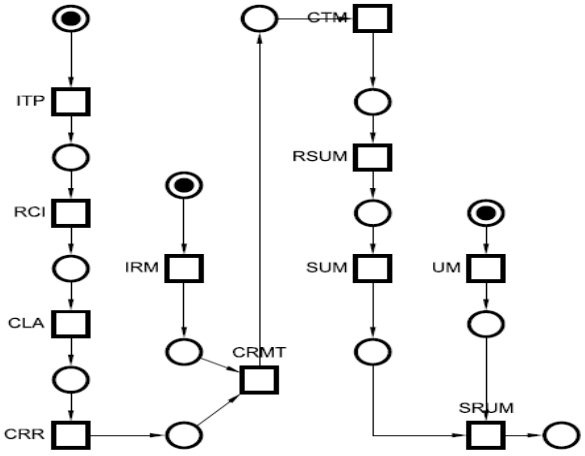

Fig 4 .PN model of AFL system Architecture

3.2 Client Side PN modeling of asynchronous Federated Learning

Clients first initialize the configuration through Initialize client configuration (IUCC) and initialize the model through (IMC).Once the client becomes ready, it will request token to the server. Server Authenticate client and acknowledge it and send token to the client. Received token will be activated and client request the model. Based on the required number of request of client, server will broadcast the model .Client will send the signal received the server model. Client builds the model and raise the signal (CBM) client build model and it also restore the session. Client will train model as per required epoch. After completion of training model will be pushed and send the signal for indication of completion of training to the server.

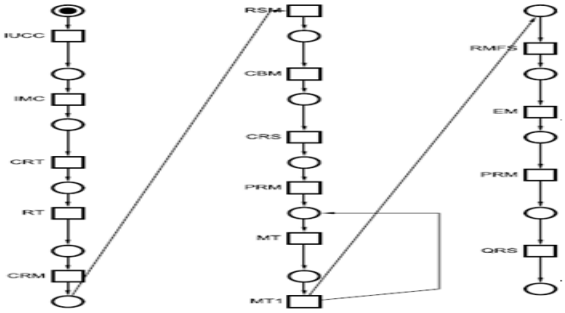

Fig 5. PN model of Client side AFL

3.3 Server side PN modeling of asynchronous Federated Learning

When starting a federated learning, server side services configure the files by raising initialize server configure file signal (ISCF) and initialize the model variable (IMV) high. After this server enters into loop, waiting for the client's joining request. During this time duration it also detects the ill posed client. Once the server found the required number of client, it will broad cast the model for the client. Server waits for the request for sending the updated model from client. After receiving the required number of updated models form client, then server starts the Aggregation process.

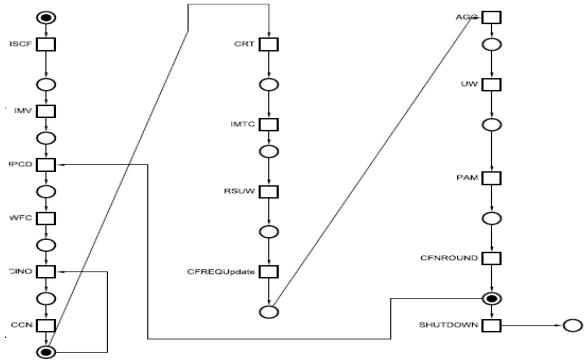

Fig 6 PN model of server side AFL

4.1 NVIDIA FLARE

NVIDIA FLARE is NVIDIA Federated Learning Application Runtime Environment. It is a installable package of group of development tools. It is python Library. It is designed to facilitate fedral learning among different client. It provides protection of data due to client side

training. It can coordinate and exchange the resulting model .It provides the data privacy. As shown in fig 7 It consists of FL simulator, learning algorithem,federation workflow, Flare programming API, Privacy preservation and secure management and tools .NVIDIA FLARE includes components that allow researchers and developers to build and deploy end-to-end federated learning applications.

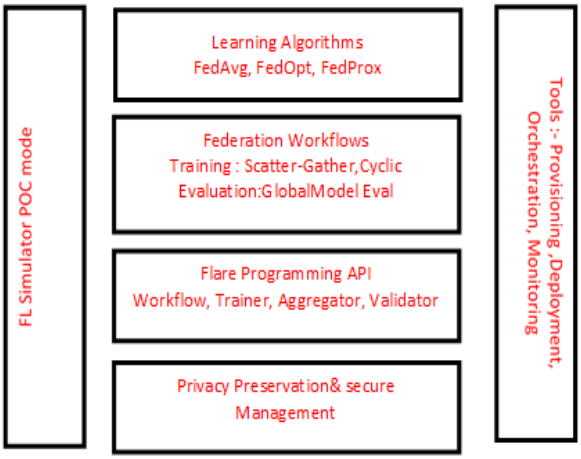

Fig 7 The high-level NVIDIA FLARE architecture

FedAvg is used for federated aggregation and equal distribution of model parameters for every local model. FedProx can be viewed as a generalization and re-parametrization of FedAvg. It adds a regularizer to the loss. FedOpt used to promote communication efficiency and privacy preservation in FL. It uses a new Shareable Generator to update the global model on the server using a PyTorch optimizer. SCAFFOLD adds a correction term during local training following the implementation . The data of different parties are usually non-independently and identically distributed (i.e., non-IID). It is data partitioning strategies to cover the typical non-IID data cases. Federated Learning Simulator (FLSim) is a flexible, standalone library written in PyTorch that simulates FL settings with a minimal, easy-to-use API.

4.2 Workcraft

Workcraft tool is used for interpretation of graph model .In this software , one can very easily visualize ,edit ,analyzed and simulate model. Simulation and verification of higher level models that are easily manipulated. Modeling causality and concurrency and Designing asynchronous circuits is possible with Workcraft.

5 Experimentation and Result

Lung cancer is found in lungs and it is leading cause of cancer death in India. Lung cancer prediction system assists the people to find the probability of risk of cancer. It helps the people to take the decision on their lung cancer risk status based on person's age, breathing style, life style etc. On an Kaggle, there is lung cancer dataset with 16 attributes and 284 instances. The various features of dataset consists of gender, age, smoking, yellow finger, anxiety, peer pressure chronic diseases, fatigue, allergy, coughing, shortness of breath, swallowing difficulty, chest pain Lung cancer[12]. Classification model is made based on this dataset using classical approach with 70 % data for training and 30 % for testing. For lung cancer dataset, classification benchmark we report Average F1 score, precision, recall and balanced accuracy based on classical model of Machine learning. For synchronous federated Learning, lung cancer dataset is spliced into three parts and placed in three different folders. With NVIDIA flare framework with one server and three client are created .The model is initialized on the server and send to three client. The updated model from three client are averaged with FedAvg algorithms. The performance of synchronous Fedrated learning is similar to the classical model of machine learning. Based on the system level behavior of NVDIA Flare, asynchronous federated learning behavior is modeled with Petri Net. Asynchronous Behavior of server side and asynchronous behavior of client is also modeled with Petri Net. The reachability analysis of all three PN model is carried out . It indicates the absence of Deadlock. There is passing of token from one place to another because transition of the signal. Asynchronous Federated learning is found in working condition in workcraft software. .Implementation of the asynchronous is required to be carried out.

6Conclusion

Asynchronous federated Learning offers distributed learning of model without sharing the data. It preserve the data privacy and security. There is straggler takes place in the synchronous federated learning as each client operated on its own speed . Two client never operates on the same speed .so the asynchronous federated learning offers better efficiency compared with synchronous federated learning as averaging takes place when required number of updated model received without waiting for all the client.

References

1) Jie Xu, Benjamin S. Glicksberg, Chang Su , PeterWalker, Jiang Bian ,FeiWang, "Federated Learning for Healthcare Informatics", Journal of Healthcare Informatics Research (2021) 5:1–19

2)Trung Kien Dang ,Xiang Lan,Jianshu Weng, Mengling Feng, "Federated Learning for Electronic Health Records", ACM Transactions on Intelligent Systems and Technology, Vol. 13, No. 5, Article 72. Publication date June 2022.

3) Prayitno , Chi-Ren Shyu , Karisma Trinanda Putra , Hsing-Chung Chen , Yuan-Yu Tsai K. S. M. Tozammel Hossain , Wei Jiang and Zon-Yin Shae , "A Systematic Review of Federated Learning in the Healthcare Area: From the Perspective of Data Properties and Applications" Appl. Sci. 2021, 11, 11191. https://doi.org/10.3390/app112311191.

4) Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics. PMLR, 1273–1282.

5)Chenhao xu, Youyang qu, Yong Xiang , and Longxiang Gao, "Asynchronous Federated Learning on Heterogeneous Devices: A Survey" ACM Comput, Vol. 37, No. 4, Article 111. Publication date: August 2021.

6) Yuhong Wen, Wenqi Li, Holger Roth and Prerna Dogra, "Federated Learning powered by NVIDIA Clara"

Technical Blog Dec 01, 2019, https://developer.nvidia.com/blog/federated-learning-clara.

7) Wenqi Li, Fausto Milletar`ı, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust1, Yan Cheng, S´ebastien Ourselin, M. Jorge Cardoso, and Andrew Feng "Privacy-preserving Federated Brain Tumour Segmentation" arXiv:1910.00962v1 [cs.CV] 2 Oct 2019.

8) Zheyi Chen , Weixian Liao , Kun Hua,, Chao Lu , Wei Yu , "Towards asynchronous federated learning for heterogeneous edge-powered internet of things", Digital Communications and Networks , Volume 7, Issue 3, August 2021, Pages 317-326.

9) Sin Kit Lo, Qinghua Lu, Liming Zhu, Hye-Young Paik, Xiwei Xu, Chen Wang, "Architectural Patterns for the Design of Federated Learning Systems" arXiv:2101.02373v3 [cs.LG] 18 Jun 2021.

10) Momina Shaheen , Muhammad Shoaib Farooq , Tariq Umer Byung-Seo Kim , "Applications of Federated Learning; Taxonomy, Challenges, and Research Trends" Electronics 2022, 11, 670. https://doi.org/10.3390/electronics11040670.

11) T. Murata, "Petri nets: Properties, analysis and applications", Proceedings of the IEEE (Volume: 77, Issue: 4, April 1989) page 541 – 580, DOI: 10.1109/5.24143

12) https://www.kaggle.com/code/michaelbryantds/lung-cancer-survey-classification/data

Biography of Authors

Prof. Vivek E. Khetade is completed PhD from Rashtrasant Tukdoji Maharaj Nagpur University, India, in the field of "Data synchronization in GALS SOC". He is an Assistant Professor in the Department of Electronic Design Technology of Ramdeobaba college of Engineering & Management. His area of interest is to design and development of Asynchronous circuit and system .His Masters is in the field of Electronic Design Technology.