COMP5434 BIG DATA COMPUTING
Group Project

Group 20
LUI MAN HON 20000963g
HUANG MIAOSHAN 20088677g
ZHANG YUANSHUO 21057033g
MA ZHE 20097946g
WANG YIQUN 21053158g
08/12/2021

Introduction

The purpose of this report is to design a privacy-protected federated credit evaluation program to predict the credit level (between 1 to 10) of customers from banks according to the given basic information, such as location, income level, and card status etc. With a view to figuring out an optimal model on the federated learning task, designing a neuron network model and evaluate its performance and impact is a steppingstone when it comes to achieve our goal.

Data preprocessing

	CustomerId	Geography	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	CreditLevel
0	15762418	Spain	3	121681.82	1	1	0	128643.35	1	8
1	15749905	Spain	6	0.00	1	1	0	50213.81	1	7
2	15600911	France	2	182888.08	1	1	0	3061.00	0	7
3	15572762	Germany	2	102278.79	2	1	0	89822.48	0	2
4	15627848	France	7	109346.13	2	1	0	102665.92	0	7
...	...	...	...	...	...	...	...	...	...	...
8995	15769645	France	3	0.00	1	1	1	48108.72	0	6
8996	15635905	Spain	6	0.00	2	1	1	43001.46	0	6
8997	15636388	Germany	7	98775.23	1	1	0	114603.96	0	7
8998	15688951	Germany	8	119654.44	2	0	1	148412.24	1	9
8999	15581229	Germany	1	173340.83	1	0	1	122763.95	0	4

9000 rows × 10 columns

Fig. 1. Original dataset

Dropping a meaningless feature Customer ID, converting the categorical feature Geography into numerical data, and normalizing the entire dataset to between 0 and 1 are our data processing steps. Since there is not any missing or incorrect data which can be found, no data cleansing job is done.

In addition, the finalized dataset is separated into 2 parts. 80% of data is the training dataset, and 20% of data is the testing dataset.

Data analytics

From the heatmap (Fig. 2), the linear correlation between each feature is extremely low, especially the column of credit level. Therefore, we decide not to drop any other columns apart from the Customer ID.

Fig. 2. Heatmap

Model design and implementation

To design a neuron network model, refer to the rule-of-thumb method for determining the correct number of neurons to use in the hidden layers, the number of hidden neurons should be less than 2 times the sum of size of the input layer and size of the output layer. Therefore, our 1st draft neuron network model is designed to have 1 input layer with 8 neurons, 2 hidden layers in total (1 hidden layer with 16 neurons, and 1 hidden layer with 8 neurons), 2 dropout layers with value 0.9 behind each hidden layer, and 1 output layer with 10 neurons. The activation function is ReLU function. The cost function is cross entropy loss function. The optimizer is stochastic gradient descent (SGD) with momentum 0.9 and learning rate 0.01. From the prediction result in Fig. 3, the testing accuracy is around 0.205 to 0.210. However, from the observation in Fig. 4, the prediction values are always 5 (class 6) after the 1st model is trained.

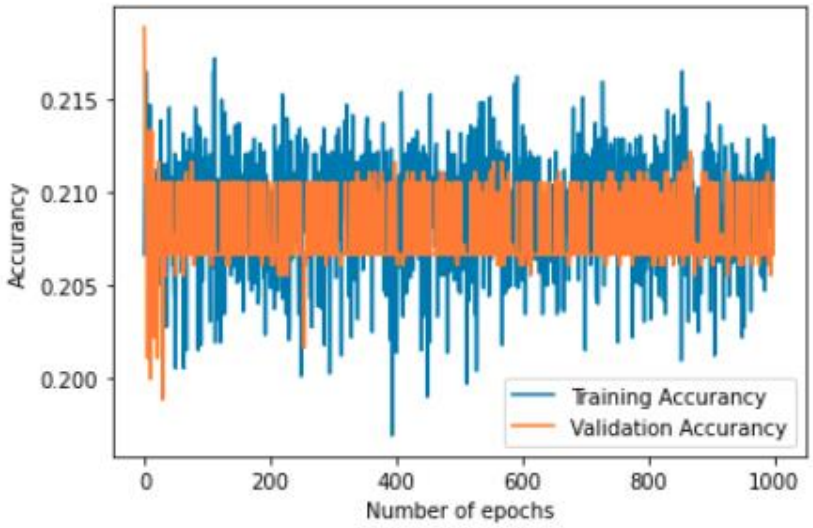

Fig. 3. Accuracy of the 1st draft model

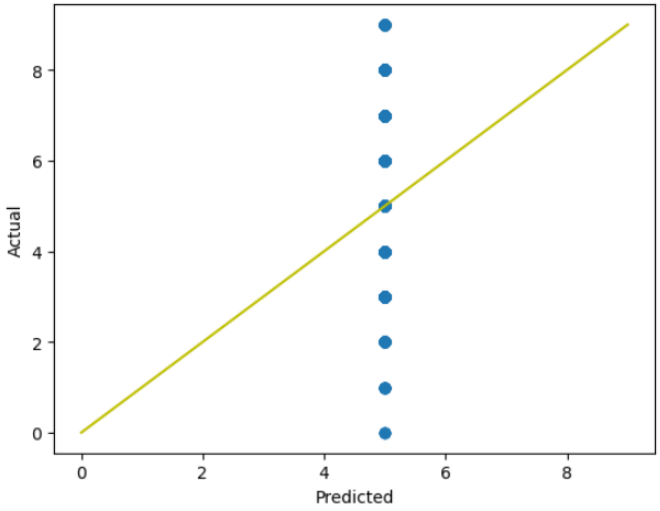

Fig. 4. Comparison between the actual and predicted value

To enhance the performance, trying different learning rate in SGD optimizer is the first step. From the prediction result in Fig. 5, for the training accuracy, the model obtains the best performance when learning rate is equal to 0.005. However, even though both training and testing accuracy are coverage after the number of epochs is larger than 2000, they are closer with one another between different value of learning rate.

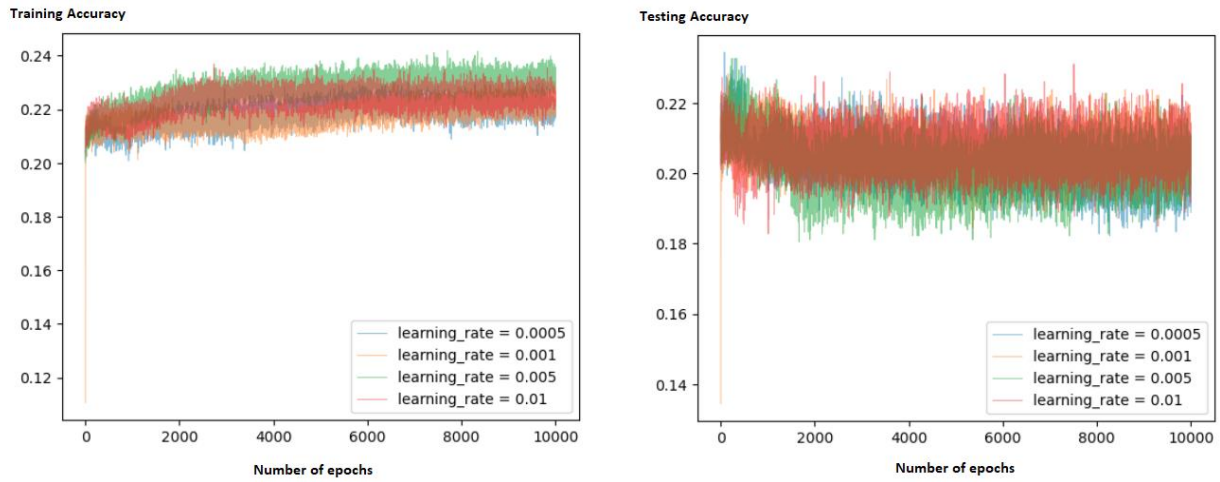

Fig. 5. Accuracy with different learning rate

In Fig. 6, by fixing the learning rate to 0.001 and testing different values of momentum in SGD optimizer, both training and testing accuracy have slightly increasing trend while the number of epochs increases, and the momentum is equal to 0.75.

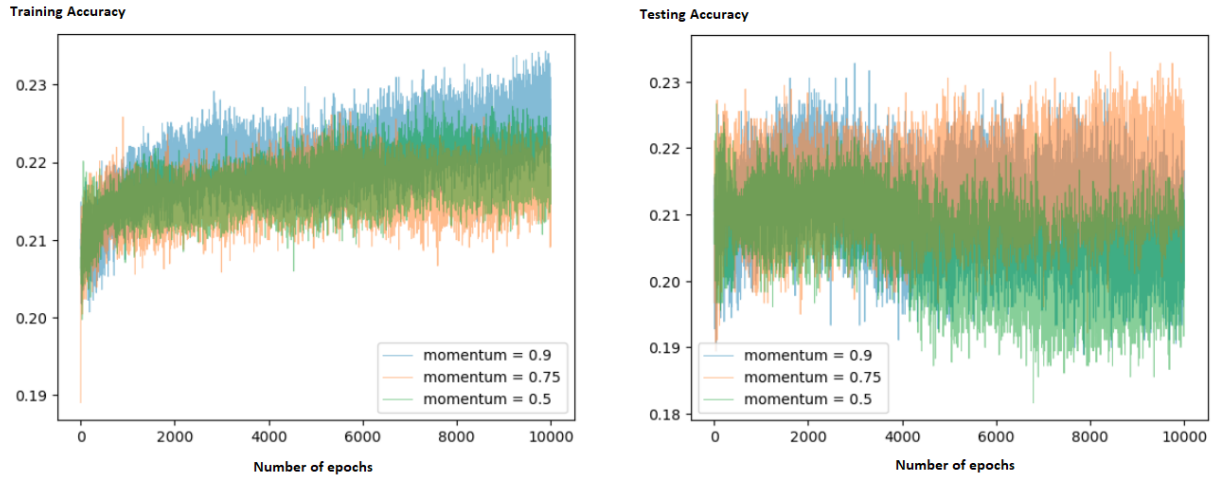

Fig. 6. Accuracy with different momentum

To continue the next experiment, in Fig. 7, by fixing the hyperparameters in SGD optimizer with learning rate = 0.001 and momentum = 0.75, and using different values in dropout layers, all testing accuracy is coverage to a similar accuracy value around 0.21, and meanwhile the training accuracy has better performance when the dropout value is equal to 0.1.

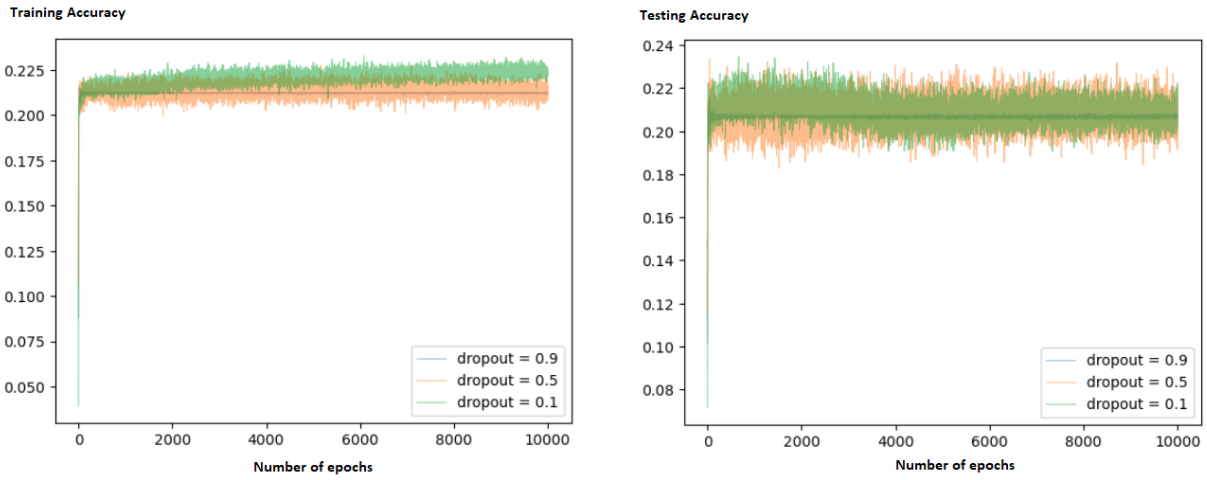

Fig. 7. Accuracy with different value in dropout layers

After the above experiments, our final neuron network model uses the same neuron network structure as the 1st model, but the dropout value is 0.1 and the SGD optimizer uses momentum = 0.75 and learning rate = 0.001. From the prediction result in Fig. 8, the testing accuracy is around 0.20 to 0.22 which has slight improvement 0.01 comparing with the 1st model. Besides, the prediction values in Fig 9 are no longer 5 (class 6) or 6 (class 7) only after the model is trained for around 20000 number of epochs.

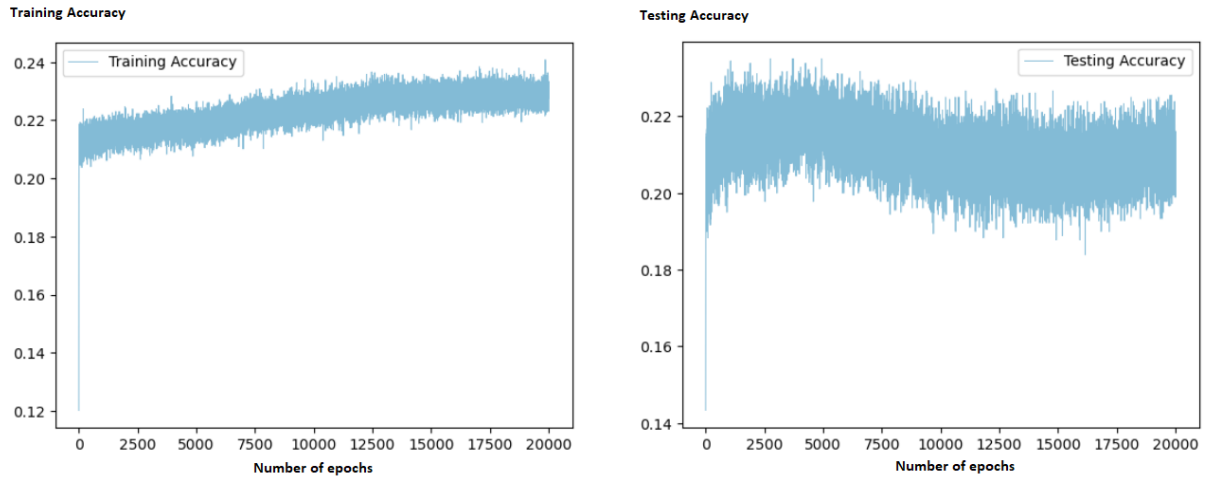

Fig. 8. Accuracy of the final model

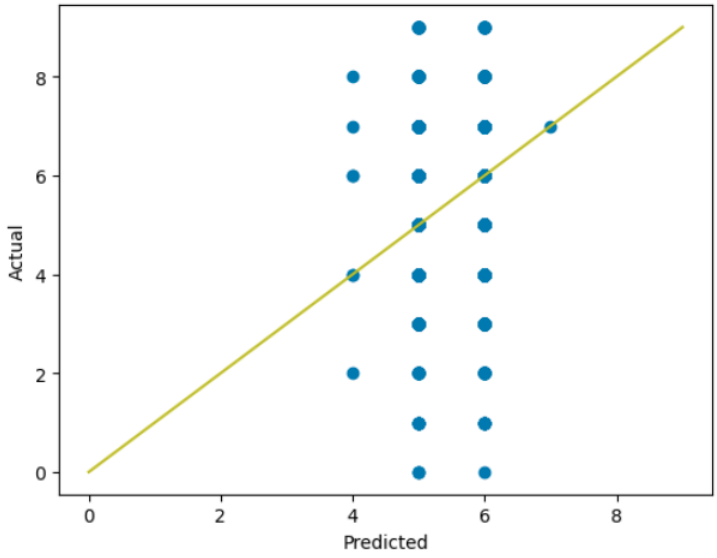

Fig. 9. Comparison between the actual and predicted value

Framework of federated learning

After the neuron network model is designed, the next step is to apply it into federated learning. Using equally random distribution to partition the training dataset into 5 parties to simulate the independent and identically distributed (iid) data between the parties is our first job. The second job is to use the Dirichlet distribution with alpha = 0.1 to partition the training dataset into 5 parties to simulate the non-iid data between the parties. The purpose is to evaluate the influence of non-iid data partitioning, since in the real-world, the data in federated learning is usually non-iid.

In Fig. 10, there is the distribution of the simulation result after the non-iid data partitioning.

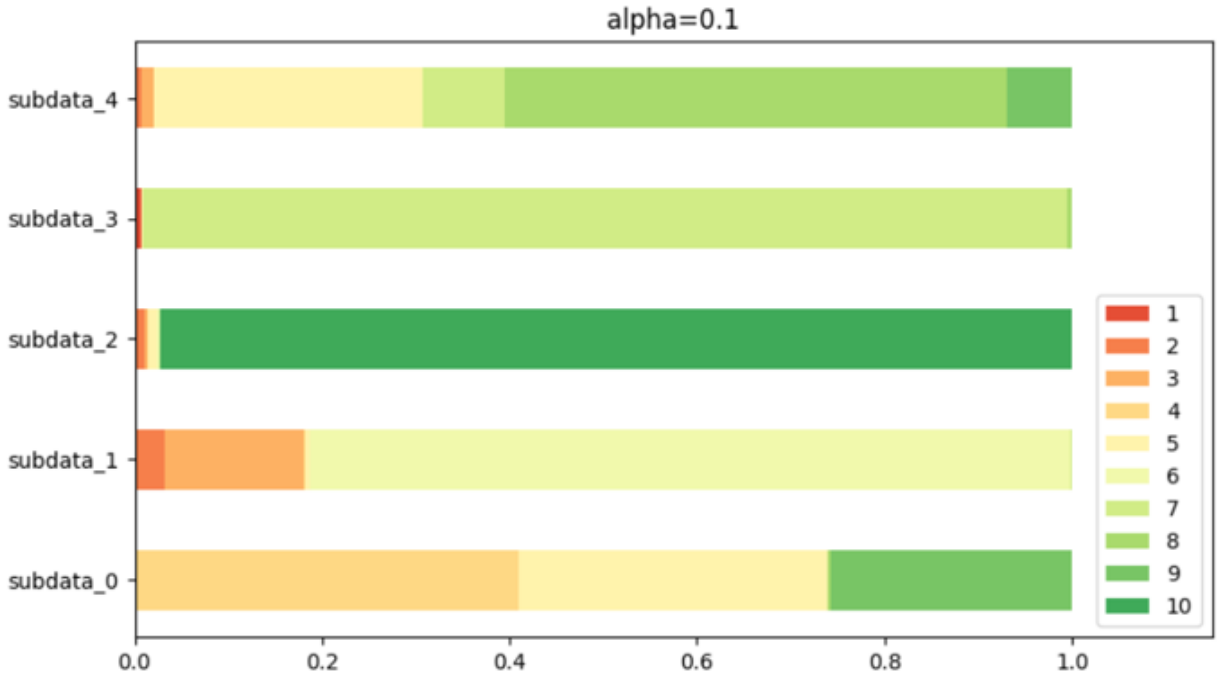

Fig. 10. Distribution of 5 parties after the Dirichlet partitioning

After the data partitioning, the aggregator takes advantage of FedAvg algorithm to aggregate the weight parameters from each party. We take the global rounds be 20.

From observation in Fig. 11, the superiority of collaborative training over individual local training is that the global accuracy is much more stable than local accuracy for each party even if the number of global rounds is increasing. The trend is almost a constant straight line in high probability.

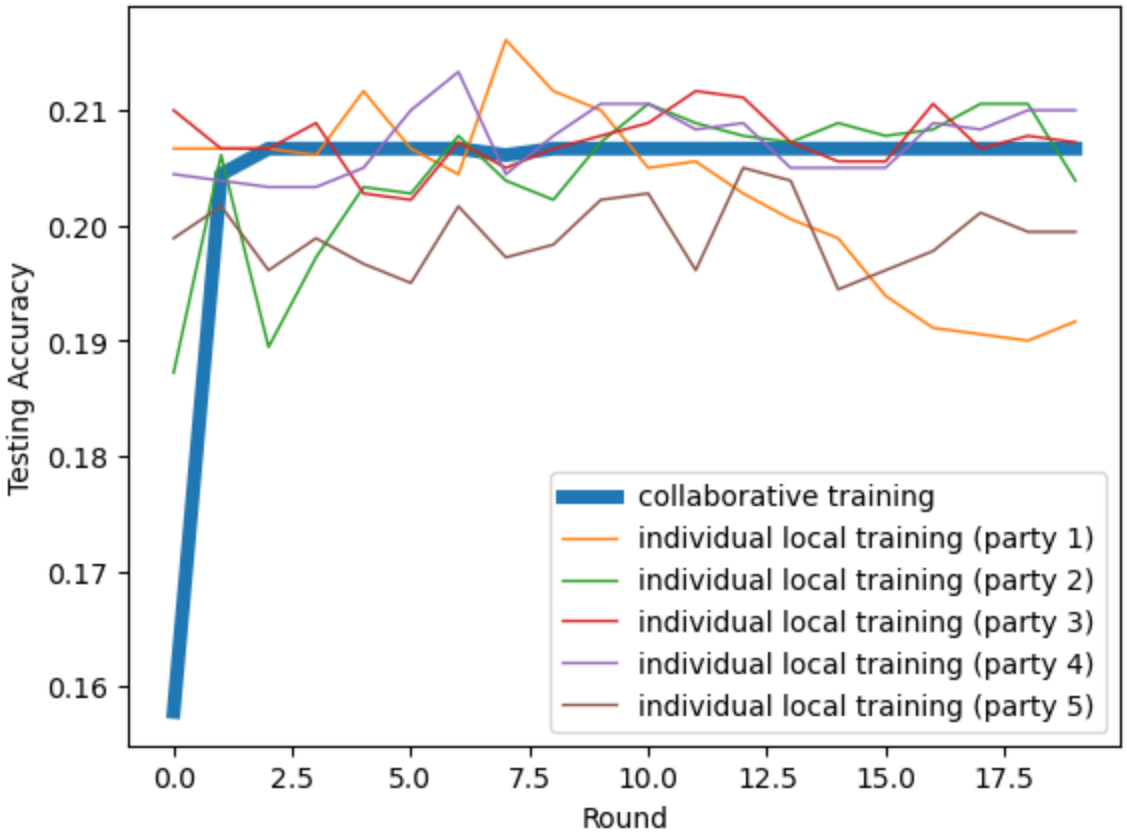

Fig. 11. Difference between collaborative training and individual local training

However, there is a challenge on federated learning. In Fig. 12, after 100 number of attempts, we re-train and re-test both two models. In comparison of the models using iid data and non-iid data, iid model has a much more stable prediction accuracy. Almost more than 95% of prediction accuracy is between 0.20 and 0.21. In contrast, the prediction accuracy of the non-iid model is much more unstable and its variance is relatively high.

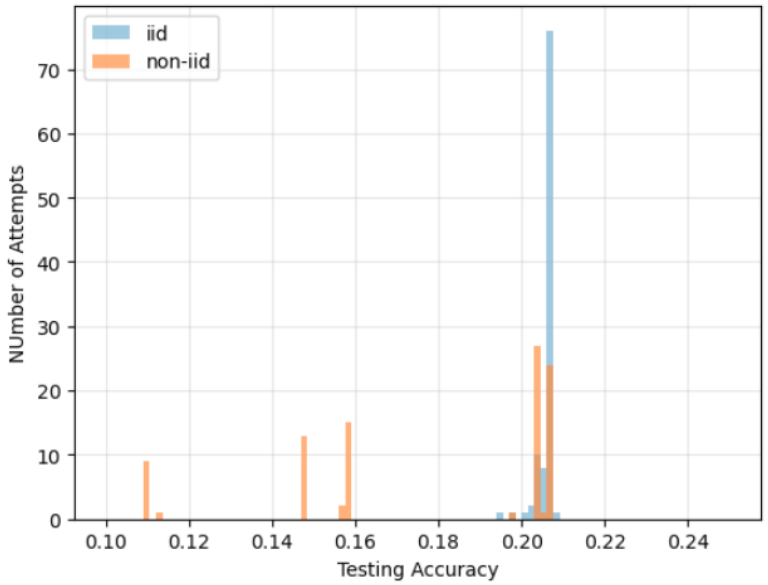

Fig. 12. Comparison of accuracy between iid and non-iid model

Performance evaluation and discussions

To achieve higher stability, we make a simple implementation on the global aggregation process. For each global round, after running FedAvg algorithm and testing the global accuracy, we add a logic that if the new global testing accuracy is dropping more than 25% from the last global round, the weight parameter update of this round will be rollback. After using this approach, the instability issue has slightly improvement shown in Fig. 13. The variance of non-iid model decreases 10% from 0.0019 to 0.0017. Meanwhile, the mean of the testing accuracy of non-iid model increases from 0.1705 to 0.1708. By tunning the percentage variable in the condition, we can get a better result.

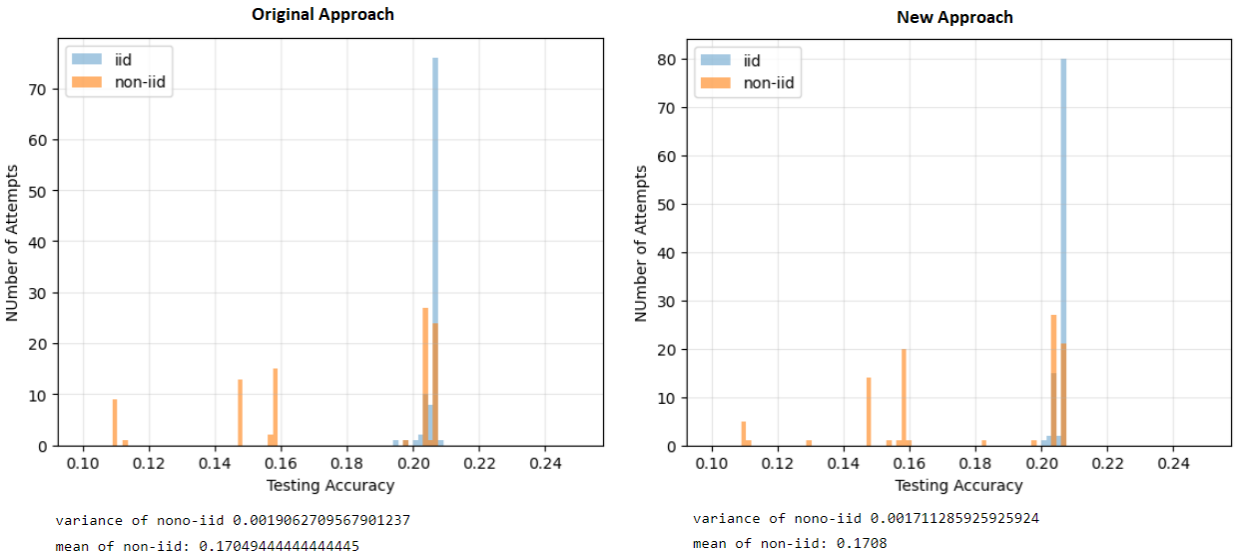

Fig. 13. Comparison of accuracy between iid and non-iid model

Summary and future work

This approach to handle the non-iid data does work to achieve higher stability even though the designed neuron network model may not be the best one. In the future, we will continue improve the NN model structure such as changing the number of hidden layers or neurons, and will test different kinds of algorithm like SCAFFOLD, FedNova, FedProx, and FAVOR to replace the current using algorithm FedAvg in order to obtain a much more stable and higher prediction accuracy result when dealing with the non-iid data in federated learning.

Reference

Kim-seng CHIA, Herlina ABDUL RAHIM, Ruzairi ABDUL RAHIM. (2011). Technical report: Neural network and principal component regression in non-destructive soluble solids content assessment: A comparison
https://www.researchgate.net/publication/221803158_Technical_report_Neural_network_and_principal_component_regression_in_non-destructive_soluble_solids_content_assessment_A_comparison

Brendan McMahan and Daniel Ramage. (2017). Federated Learning: Collaborative Machine Learning without Centralized Training Data
https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, Vikas Chandra. (2018).

Federated Learning with Non-IID Data
https://arxiv.org/pdf/1806.00582.pdf

Hangyu Zhu, Jinjin Xu, Shiqing Liu, Yaochu Jin. (2021). Federated Learning on Non-IID Data: A Survey
https://arxiv.org/pdf/2106.06843.pdf

Daniel Cotto. (2019). Neural Networks for Option Pricing
https://towardsdatascience.com/neural-networks-for-option-pricing-danielcotto-c24569ad0bb