Federated Learning for Image Classification

S M Sarwar[1], Dr. Honglu Jiang[2]
[1] Department of Computer Science, [2] Department of Cyberspace
College of Engineering & Computer Science

ABSTRACT

This work contains the steps followed to replicate and improve a federated learning classification algorithm for the MNIST data set. A brief description of what federated learning means and how it is different from traditional machine learning is included, along with the methodology used and code for the essential parts of the algorithm. A total of 18 experiments are performed. These experiments come from modifying settings such as the number of epochs, batch size, number of clients, and number of training rounds. Finally, an analysis comparing the results obtained is included, and future work is discussed.

BACKGROUND

We know the performance of machine learning models depends on the data used to train the model. Therefore, these algorithms need to be fed with a large amount of representative data to achieve higher accuracy. As we become more and more dependent on our devices, such as smartphones and computers, the information we generate increases with such dependability. As the consumers of devices and applications, we are the ones generating the data required to train the machine learning models that make predictions. However, we often need to share our private, sensitive information with corporations, which creates a security concern regarding our information.

We asked ourselves for this project: How can we train a machine learning model using user-sensitive information without compromising users' privacy? The goal of this project is to implement a machine learning algorithm that could address the following two main points:

- Data Privacy: Data is shared with corporations, and we can just trust that they will not misuse our sensitive, private information and that hackers will not steal it.
- Data Availability: The amount of information generated is enormous compared to the actual data we access. There has to be an easier way to "collect" data representing the population.

In traditional machine learning, we train a model at a centralized server with some data which is also collected from the user. The performance of machine learning models we know depends on the data used to train the model. We, as the consumers of devices and applications, are the ones generating the data that is required to train the machine learning models for predictions. However, many times we need to share our private, sensitive information with corporations, and this creates a security concern regarding users' private information. But at Federated learning (FL) , it is possible to train a model without sharing user's data. FL has been introduced by H. Brendan McMahan, Eider Moore from Google in 2017 [1]. FL provides a new facility for user privacy by training data on users' (client) devices and passing that trained models or parameters to the central server without sharing users' data.

FL is a decentralized form of machine learning, where the model is trained on the clients' devices themselves rather than at a centralized server. For this type of architecture, a federated data set is required. That is, data will not be shared with corporations, and the owner of the model will only receive parameters after the training is performed by each user on their own devices. With this procedure, the information is not vulnerable because it is not shared, and the model will still receive relevant information for the machine learning algorithm.

In this paper, we use the same scenario as the base of our project and show how to train a federated learning model using the MNIST data set.

ADVANTAGES

Many practicalities come with using FL. This new machine learning technology improves user privacy and protection, but it improves latency issues associated with traditional machine learning methods. Additionally, FL provides a hyper-personalized model for the user since it uses the client's data directly without actually using the information itself.

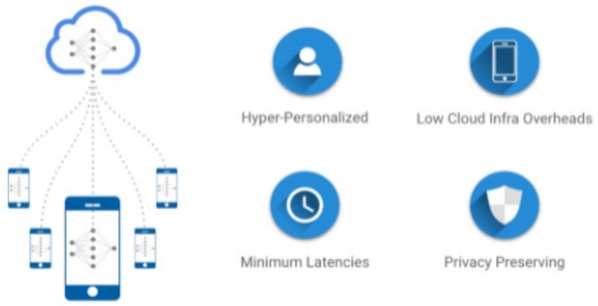

Fig. 1: Advantages of Federated Learning

METHODOLOGY

Note that the data itself would not be readily available in an actual scenario as FL aims to maintain user privacy. However, in this practical example, the MNIST data is used locally to track the progress of our FL algorithm and can be distributed to the users evenly at our discretion.

For the framework used, we have the following: TensorFlow Federated (TFF) Initial Public Release: Feb 19, 2019 Current Version: 0.19.0 Author: Google Inc. Type: Open-Source

First, we divided the given data set into two parts: True and False.
only_digits= True: 3,383 users, 10 label classes. We then divided again into two parts labeled as train: 341,873 examples and test: 40,832 examples.
only_digits= False: 3,400 users, 62 label classes

We divided it into two parts as train: 671,585 examples and test: 77,483 examples.

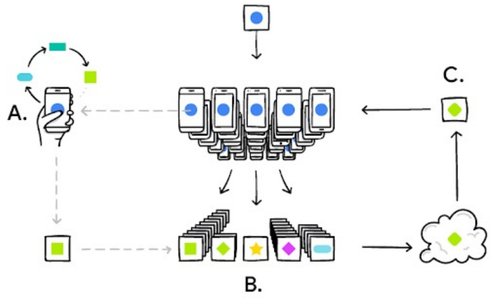

Fig. 2: Flow Chart for Federated Learning

To illustrate a simple FL model, the reader can refer to Fig. 2. Here, the developer initially deploys a model for multiple clients. Then those clients begin training the model in the background on their personal devices whenever they are not being used. Over a certain period, the training done on the users' devices will be uploaded to a central server where the model can be refined. Finally, again the developer will send out the new and improved model and repeat the process where they will analyze updates and release a new update/version to all the users.

RESULTS & DISCUSSIONS

This section is used to showcase the results obtained from the eighteen experiments that were performed throughout the course of this project. The results varied as we experimented with the different number of epochs, training rounds, batch size, and the number of users. These parameters were used and modified because they were the most straightforward to interpret and provided a basis for future experimentation. The table labeled "Summary of Experiments" summarizes the different parameters used for each experiment.

Loss Plot: 2.166

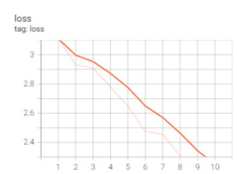

Fig 3. Loss Plot

Accuracy Plot: 0.2992

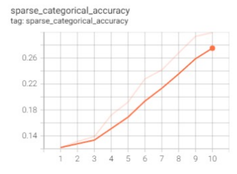

Fig 4. Accuracy Plot

The plots shown in Fig. 3 and Fig. 4 represent the initial data provided after it has been processed given these parameters: 5 epochs, 11 training rounds, batch size of 8, 10 users. As shown, the loss plot stopped at about 2.166 while the accuracy plot went up to only 0.2992. This was only a result of having such a small amount of training rounds, epochs, etc. Here the goal was to improve these outcomes.

CONCLUSIONS & FUTURE WORKS

As the first project of Fl, we tried to implement by TensorFlow Federated framework, and it was just a primary implementation on FL MNIST Dataset. As a result of the previously mentioned constraints, we could only work with the basic parameters that we felt were the simplest to modify and understand the reasoning behind the given outcome. we will soon start working with different FL architectures and different types of privacy algorithm applications on the different architectures. Also, we will try to improve accuracy and privacy.

ACKNOWLEDGMENTS

We would like to thank Cristian Vega from the College of Engineering & Computer Science and Dennis Rodriguez from the School of Mathematics & Statistical Sciences for their help for this project. Also, We would like to mention the name of Dr. Dong-Chul Kim from dept. Of Computer Science for his valuable advice.

REFERENCES

1. H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson and Blaise Ag¨uera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data, 2016, Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017. JMLR: W&CP volume 54; arXiv:1602.05629
2. Google Fl Team, https://ai.googleblog.com/2017/04/federated-learning-collaborative.html