Comparative Analysis of Federated and Non-federated CNN for Image Classification

Anika Tasnim
Department of CSE
BRAC University
Dhaka, Bangladesh
tasnimanika59@gmail.com

Md. Sajeebul Islam Sk.
Department of CSE
BRAC University
Dhaka, Bangladesh
sajeebul.islam.sk@g.bracu.ac.bd

Zarin Anan Aunshu
Department of CSE
BRAC University
Dhaka, Bangladesh
zarin.anan.aunshu@g.bracu.ac.bd

***Abstract*—Federated learning allows multiple parties to train a machine learning model collaboratively without sharing their local data. Handling the heterogeneity of local data distribution across parties is a crucial challenge in federated learning. Federated learning offers secure models without data sharing, resulting in a highly effective privacy-preserving approach that both offers security and data access. Despite the fact that many studies have been proposed to address this issue, we find that deep learning models fail to achieve high performance in image datasets. In this research, we have analyzed the performance of federated and non-federated CNNs. When testing non-federated CNN, we are able to reach 56% accuracy, while federated CNN is tested on the server we have achieved 59.62% accuracy. The CIFAR10 dataset has been used to evaluate our models.**

***Index Terms*—CNN, pytorch, CIFER10, Machine Learning, Federated Models, Federated CNN etc.**

I. Introduction

A machine learning (ML) approach called federated learning (FL) enables distributed, edge devices to train an ML model cooperatively without sharing data. FL has been thoroughly investigated in a variety of confidentiality application fields in recent times. The accessibility of high-quality annotated data at each participating client to effectively train local models places further restrictions on the viability of FL in an application. One of the key bottlenecks is the data collecting and annotation process, particularly in supervised ML where human annotators are typically utilized to annotate training data for ML models. To manually examine and annotate data, a sizable community participates in a crowd-sourcing activity. There are two main obstacles in the procedure. First and foremost, each sample must be properly examined, which is a laborious and time-consuming task. Second, the approach does not ensure that high-quality samples are chosen, which would make the model's performance more relevant and impactful [2]. FL and distributed learning have a close relationship. Distributed computing and distributed storage make up a traditional distributed system. Although FL placed a lot of focus on privacy preservation, the most recent research in distributed machine learning also gives privacy-preserving distributed systems a lot of attention. Through the use of a communication network, numerous computers in various places may be connected and managed by a central server to perform distributed processing, in which each computer completes various components of a single task. As a result, FL concentrates on developing a collaborative model without privacy leaks whereas distributed processing is primarily focused on expediting the processing step [4].

T. Anika, S. Hasan and Z.A. Aunshu was with the Department of Computer Science and Engineering, BRAC University, 66 Mohakhali, Dhaka, Bangladesh

Feature extraction techniques that have been successfully tested for a variety of visual recognition tests are frequently employed for image representation in classification challenges. The majority of implemented features need professional identification and hand coding according to the data type and domain. This procedure is challenging and time- and money-consuming. By automating the procedure of extracting and learning features, deep learning decreases the burden of creating new feature extractors. Utilizing this technology, the proposed traffic sign classification system is able to identify the pictures of traffic signs placed on the road and categorize them. There are several deep learning architectures available. CNNs, the most effective and practical deep neural network for this sort of data, were utilized to create the model reported in this research, a classifier system. CNNs that have been trained on huge datasets of pictures for recognition tasks may thus be made more effective by applying these learning representations to tasks that only require a small amount of training data [3]. The design of convolutional layer and pooling layer, the activation function, the loss function, regularization, and the applicability of CNN to actual applications are the key areas of focus for convolutional neural network optimization [1].

Inspired by their work, we apply both model for same dataset (cifar-10) and discuss about their performance and correctness in light of their work.

In this paper, we will explain the background and related works in section II. The used dataset is explained in section III with it's description. The methodology that we are planning to use is explained in section IV. In section V the methodology is implemented and results are discussed followed by a conclusion and proper reference and acknowledgements.

II. Background and Literature Review

[2] examines how FL might profit from unlabeled data made available by AL at each participating client. In order to do this, they use and assess a number of AL approaches in two distinct application areas to propose an AL-based FL framework. They demonstrate that AL is equally beneficial in federated and centralized learning through a complex experimental setting by attaining equivalent results with manually labeled data utilizing fewer samples without requiring human annotators in training data collection. By assessing the suggested technique in two intriguing applications—waste categorization and natural disaster analysis—each with their own unique characteristics and difficulties, they also showed that the method is dataset/application agnostic.

Convolutional neural networks [3],[1] have shown to perform well when classifying images. A straightforward Convolutional neural network for image categorization was built in [1]. The picture categorization was accomplished using this straightforward Convolutional neural network. On the benchmarking datasets minist and cifar-10, their experiments are based. They also investigated several approaches to setting the learning rate and various optimization algorithms for resolving the best parameters that impact picture categorization on the basis of the convolutional neural network.

Federated learning [4],[5] entails localized data retention while training statistical models over distant devices or solid data centers, such as mobile phones or hospitals. A major change from the accepted methods for large-scale machine learning, distributed optimization, and privacy-preserving data analysis is required when training in diverse and potentially huge networks. In [5], they explain the distinctive features and difficulties of federated learning, give a comprehensive assessment of existing methodologies, and identify numerous future research objectives that are pertinent to many different research fields.

The purpose of [4] is to analyze current industrial engineering applications in order to provide guidance for upcoming landing applications. Additionally, their analysis proposes six research areas that will focus on FL literature and aid in understanding FL for potential future optimization. Their research aids in the conclusion of applications in computer science and industrial engineering and summarizes a survey of applications in Florida.

A learning strategy using convolutional neural networks (CNN) training for a traffic sign categorization system is described in [3]. The preliminary classification results of using this CNN to train features and categorize RGB-D pictures job are also shown. They investigate the transfer learning method known as the ”fine tuning methodology,” which involves leveraging layers that have been trained on the ImageNet dataset to produce a solution for a four-class classification job on a new batch of data.

III. Dataset

For this work, we are using the CIFER10 dataset. A collection of images known as the CIFAR-10 dataset is frequently used to train computer vision and machine learning technologies. One of the most popular sets of data for machine learning studies is this one. 60,000, 32x32 color images in 10 distinct categories make up the CIFAR-10 dataset. The ten categories include trucks, frogs, horses, deer, dogs, cats, birds, cats, ships, and cars. Each class has 6,000 photos in total. Object recognition algorithms in computers frequently pick up new skills through practice. A collection of photos called CIFAR-10 can be used to train a computer to recognize items. The low-resolution which is 32x32 images in CIFAR-10 make it possible for previous studies to swiftly test out various methods to determine what works. CIFAR-10 is a tagged subset of the 80 million tiny pictures dataset. Students were paid to label each image in the dataset when it was made[7].

We load the dataset using the Keras API. After this we divide the overall dataset in test and train sets to run in the CNN and federated CNN to compare the two types of model to see which works better and gives a more accurate accuracy. There we compare between the normal CNN and the federated version of CNN with the same dataset. As we aldready know how the images are stored in the dataset, preprocessing the data becomes easier. For instance, we are aware that the photos are all pre-segmented, have the same square size of 32 by 32 pixels, are all in color, and each image contains a single item. As a result, we can practically quickly import the images and start modeling with them. The 10 classes are labeled from 0 to 9, in a list ’plane’, ’car’, ’bird’, ’cat’, ’deer’, ’dog’, ’frog’, ’horse’, ’ship’, and ’truck’. We prepared and imported all the necessary libraries before preprocessing and dividing the data. After the dataset is prepared, we feed the dataset into a CNN model first and then into a federated CNN model for comparison.

IV. Methodology

A. CNN

A type of artificial neural network called a convolutional neural network, or CNN, is frequently used to evaluate visual representations. Because of the shared-weight architecture of the convolution kernels or filters that slide along input features and produce translation-equivariant outputs known as feature maps, CNNs are also known as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN). Contrary to popular belief, most convolutional neural networks do not translate invariantly because of the downsampling operation they perform on the input. An input layer, hidden layers, and an output layer make up a convolutional neural network. Any middle layers in a feed-forward neural network are referred to as hidden layers since the activation function and final convolution hide their inputs and outputs. The hidden layers in a convolutional neural network contain convolutional layers. This typically contains a layer that does a dot product of the input matrix of the layer with the convolution kernel. The activation mechanism for this product, which is often the Frobenius inner product, is frequently ReLU. The convolution procedure develops a feature map as the convolution kernel

moves across the input matrix for the layer, adding to the input of the following layer. Following this are further layers like normalizing, pooling, and fully connected layers[8].

To use CNN in a ML project we need to download the dataset. Understand the dataset to preprocess the dataset. After preprocessing, we have to load it in a way to read the labels and make it displayable. We have to help the data to be handled. Then run the dataset in a taining model which is in our case, CNN. We have to apply to the steps of CNN and all the functions and then go through each layers like maxpooling, convolution etc. [6].

B. PyTorch

PyTorch is a tensor library that has been tuned for deep learning with GPUs and CPUs. A machine learning framework called PyTorch is based on the Torch library and is used for tasks like computer vision and natural language processing. On top of PyTorch, several pieces of deep learning software are created. In our work, we use it for image classification and feature extraction. Better visualization provided by TensorFlow enables developers to more effectively debug and monitor the training process. However, PyTorch offers faster performance. TensorFlow cannot handle low-performance models like prototypes as quickly as PyTorch can. In such circumstances, it can improve outcomes.

C. Federated CNN

Federated learning, sometimes referred to as collaborative learning, is a machine learning method that utilizes a number of distributed edge devices or servers that keep local data samples to train an algorithm without transferring the data samples. This method differs from more typical decentralized techniques, which frequently presume that local data sets are uniformly distributed, as well as traditional centralized methodologies, where all local datasets are uploaded to a single server. Federated learning allows several players to develop an identical, reliable machine learning model without sharing data, enabling for the resolution of crucial concerns such data privacy, security, access rights, and heterogeneous data availability. Defense, telecommunications, internet of things, and pharmaceutical industries are just a few of the sectors where it has uses.

We want to implement the federated version of CNN to compare the results and version of CNN. For this we will distribute IDs and create multiple clients. All clients will be stored in a list and we will distribute the dataset. We will run the training model on random clients. Overall, with various epochs we will observe how the model works and over how many epochs it gives consistence results. Once we identify the consistency in results we will keep note of the epochs count to train the model in better atmosphere. The main reason of federated CNN is to see if it performed better that the normal CNN or not. If the model performs better then by how much it works well.

V. Implementation and Results

The performance of a model is analyzed and interpreted using evaluation metrics. This also shows how well or poorly the models perform. Moreover, evaluation metrics are necessary for comparing various models. Performance can be measured using a variety of metrics. Accuracy and confusion matrix are two of the performance metrics we have used to evaluate our models. In classification problems, accuracy is used to represent the percentage of correct predictions made by a model. In machine learning, the accuracy score is a metric that compares the number of correct predictions made by a model to the total number of predictions made. A Confusion matrix is a N x N matrix that is used to assess the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values to the machine learning model predictions. This provides us with a comprehensive picture of how well our classification model is performing and the types of errors it makes. Here Fig. 1 is the confusion matrix of non-federated or basic CNN. Whereas fig. 2 is the confusion matrix of federated CNN.

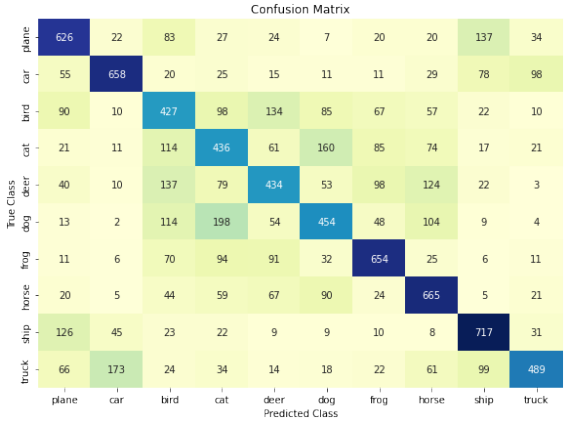

Fig. 1: Confusion Matrix of Basic CNN.

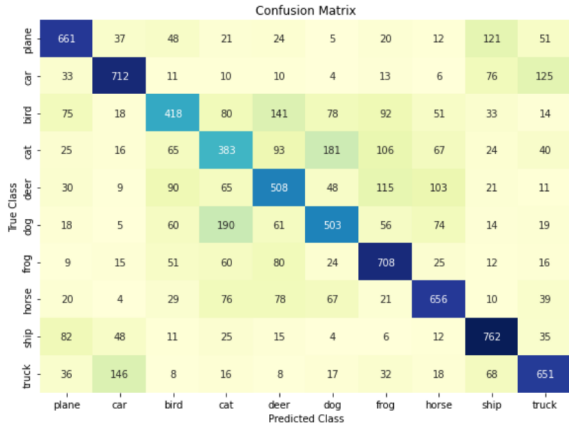

Fig. 2: Confusion Matrix of Federated CNN.

The Training loss vs. Number of epochs plot of basic CNN is shown in Figure 3. The figure shows that as the number

of epochs increases, the training loss decreases. Where Fig. 4 shows the Training accuracy vs. Number of Epochs plot of a non-federated CNN. It is a rising graph because accuracy rises with increasing epochs. From the graph we can see that, after 10 epoch, the accuracy is 56%.

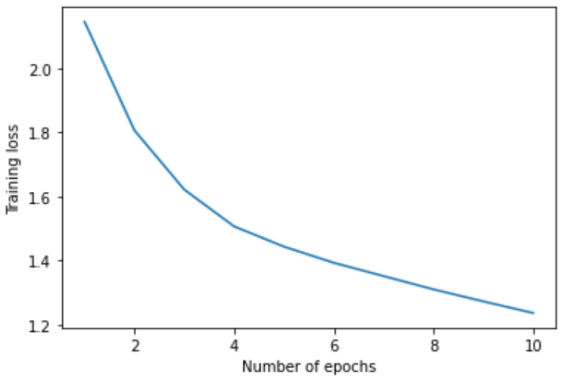

Fig. 3: Training loss vs Number of epoch plot of basic CNN.

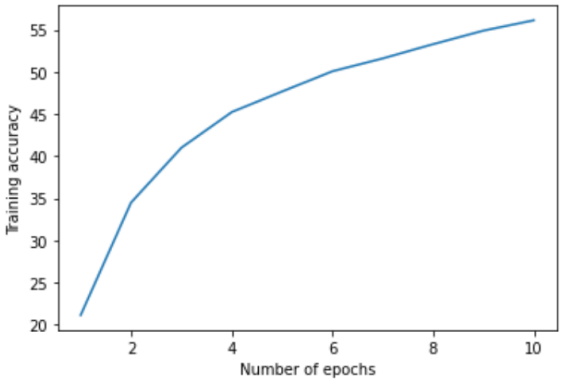

Fig. 4: Training accuracy vs. Number of Epochs plot of a non-federated CNN.

Both the Training loss vs. Number of epoch plot and the Training accuracy vs. Number of epoch plot for Federated CNN are shown in Fig. 5. According to the graph for Training loss vs Number of epochs, the testing loss is greater than 2.0 at the start but decreases to about 1.0 after 10 epochs, which is quite good. In the Training Accuracy vs. Number of Epochs graph, the accuracy rises to nearly 60%.

Table 1 shows Epoch No., Average Clients Loss, Average Clients Accuracy, Server Testing Accuracy of the federated CNN model. From the table we can see that after 1st epoch the Average Clients Loss is 0.98, the Average Clients Accuracy: is 39.16% and the Server Testing Accuracy: is 31.14%. In the 2nd epoch the Average Clients Loss becomes 0.81, the Average Clients Accuracy becomes 48.52% and the Server Testing Accuracy becomes 43.72%. So after only one epoch, the Average Clients Loss reduces up to 0.17, the Average Clients Accuracy increases by 9.36% and the Server Testing Accuracy increases by 12.58%. Overall these results are

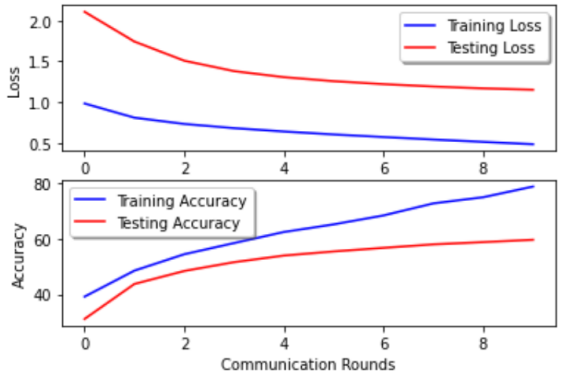

Fig. 5: Training loss vs Number of epoch plot and Training accuracy vs Number of epoch plot for Federated CNN.

really impressive. However after the 10th epoch, the Average Clients Loss becomes 0.48, Average Clients Accuracy becomes 78.76% and the Server Testing Accuracy becomes 59.62%. The data table shows that after the 10th epoch, the average server accuracy nearly doubles. This result is also superior to the basic CNN, which has a test accuracy of 56%. So, using this federated version of CNN not only provides us with better accuracy but also with data security and access.

VI. Conclusion

Federated learning (FL) is a popular method for training machine learning models on sensitive data. FL is specifically designed for training on unbalanced and non-iid data sets across participants. The latest Federated Learning approaches use differential privacy or robust aggregation to ensure the privacy and integrity of the federated model. Consequently, to put up with this factor, we have analyzed the performances of the basic CNN with the federated CNN. We achieved 56% accuracy when testing non-federated CNN for the server, and 59.62% of average accuracy when testing federated CNN on the server. The federated CNN's accuracy significantly improves with each epoch. Furthermore, the server's average loss was 1.15. So it really shows a quite good result. Furthermore, XAI can be used to improve the model's understanding. Because most deep learning models are black boxes, they cannot provide a proper explanation for their predictions. As a result, XAI will make the model more interpretable and transparent.

VII. Acknowledgement

We would like to thank our respected teacher for giving me the chance to work on this research paper as it helped me a lot to learn about this topic and increased my knowledge.

Beyond all else, I am grateful to the Great Almighty, the fountain of all wisdom and knowledge, for his endless love.

REFERENCES

[1] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*, Beijing, China: IEEE, Mar. 2017, pp. 721–724, ISBN: 9781509036189 9781509036196. DOI: 10.1109/ICBDA.2017.8078730. [Online]. Available: http://ieeexplore.ieee.org/document/8078730/ (visited on 12/20/2022).

[2] L. Ahmed, K. Ahmad, N. Said, B. Qolomany, J. Qadir, and A. Al-Fuqaha, "Active Learning Based Federated Learning for Waste and Natural Disaster Image Classification," *IEEE Access*, vol. 8, pp. 208 518–208 531, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3038676. [Online]. Available: https://ieeexplore.ieee.org/document/9261337/ (visited on 12/20/2022).

[3] S. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," en, *Neurocomputing*, vol. 378, pp. 112–119, Feb. 2020, ISSN: 09252312. DOI: 10.1016/j.neucom.2019.10.008. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231219313803 (visited on 12/20/2022).

[4] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106 854, 2020.

[5] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, May 2020, ISSN: 1053-5888, 1558-0792. DOI: 10.1109/MSP.2020.2975749. [Online]. Available: https://ieeexplore.ieee.org/document/9084352/ (visited on 12/20/2022).

[6] A. Biswal, *Convolutional neural network tutorial [update]*, Dec. 2022. [Online]. Available: https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network.

[7] *Cifar-10*, Jun. 2022. [Online]. Available: https://en.wikipedia.org/wiki/CIFAR-10.

[8] *Convolutional neural network*, Nov. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Convolutional_neural_network.