Federated Learning in Developing Countries

Maleeha Masood
Department of Computer Science
Lahore University of Management Sciences
Lahore, Pakistan
21100217@lums.edu.pk

Muhammad Nouman Abbasi
Department of Computer Science
Lahore University of Management Sciences
Lahore, Pakistan
21100177@lums.edu.pk

***Abstract*—With emerging laws like the General Data Protection Regulation (GDPR), Federated Learning is the direction where the future is heading towards. In an attempt to facilitate this cause, by trying to increase the reach of the new learning algorithm, we dive into the depths of the original state-of-the-art Federated Averaging Algorithm (FedAvg) [1] in an attempt to optimize it and make it suitable for devices with limited features and specifications- a common sight in Developing Countries. We build upon the first proposed FedAvg scheme by creating a subset of the centralized model and then assigning this new model to resource constrained devices only. Devices deemed capable of with handling the original model will receive it as it is. We conduct extensive experimentation to show that while our accuracy and convergence time is comparable to the original algorithm's accuracy and convergence times in certain circumstances, the computation load on the resource constrained devices reduces significantly. We also note a slight stability and decrease in losses with our proposed framework. We further note that reducing the number of features from half to one sixth does not have any significant effects on our results.**

Keywords— Distributed, Federated Learning, Heterogeneity, Machine Learning

I. Introduction

The journey of Federated Learning began when it was introduced by H. Brendan McMahan [1]. The main motivation behind such a technique then was the growing concerns of the privacy of individuals. Federated Averaging helped relieve this concern by introducing a framework that allowed machine learning algorithms to proceed without gathering the data of individuals [2]. This was done by sending a centrally stored model to devices that could process their on-device data using the model. Communication occurred through the transmission of parameters and at the end of this transmission, the central model updated its parameters by averaging it with the results from processing on devices.

One main concern in the original framework was the fact that not all devices would be able to spare resources for this kind of heavy processing. McMahan and other researchers at Google proposed the idea of filtering as a solution: sending the global model only to those devices that are plugged, idle and using unmetered Wi-Fi [1]. This resulted in dropouts of devices that could not keep up with the pace and resource requirements of the framework leading to the loss of processed information and crucial parameters from devices that could contribute to a more generalizable central model. Achieving high degrees of systems and statistical heterogeneity is a key problem [6] of the original state-of-the-art framework. This concern is very valid in all of its entirety: in 2017, 24% of the Android devices shipped has less than 1 GB of RAM. Only 32% of the shipped devices has greater than or equal to 4 GB of RAM. This diversity in mobile device characteristics was a feature that the Federated Algorithm was not able to reflect towards. While this problem is valid generally, however, it is alarmingly concerning in developing countries where despite having explosive mobile phone adoption statistics [3], majority of the individuals cannot afford the newest smartphone models. Given this fact and the potentially high number of dropout devices, it would be safe to say that a centralized model might rarely get any contributions and model updates from the developing regions around the World.

One of the main aims of a machine learning algorithm is to learn from a training set and then apply the learnt parameters on unseen testing data. For this, models need to be generalizable. To increase generalizability, ideally, a model should be exposed to a variety of data to train on so that it can gain a true perception of the complete environment [4]. By simply 'dropping' the data from a considerably sized fraction of the world, an unintentionally biased central model might develop, which would not predict well on unfamiliar data from developing regions.

To increase the number of contributions from developing regions, we feel that it is necessary to resize the model for devices that cannot process it due to their finite resource limitation. Building upon this, we create a subset of the original model and assign this new model to resource constrained devices only. Devices deemed capable of with handling the original model will receive it as it is. After processing, we aggregate results from all the devices and update the centralized model.

We perform extensive experimentation on our work, comparing it to the original algorithm's results at each step. We show that in certain situations, we are able to get comparable accuracies and convergence times with the original Federated Averaging Algorithm whilst reducing the strain on resource constrained devices. This strain is not just in terms of computation but also in terms of network usage since there is a significant reduction in the byte size of the total model. We also note a slight stability as well as decrease in losses with our proposed framework. Our results show that the computational time difference for devices with a subset of the model of half the number of features is up to 4 times smaller than the time required for processing by the devices that had received the complete model. We further note that reducing the number of features from half to one sixth does not have any significant effects on our results.

We do hope to build further on this work and make it competent enough to function as a substitute of the original framework. For the time being, we hope that our work successfully conveys the notion that it is possible to reduce complex models to target a greater number and diversity of users whilst maintaining the key features and properties.

The remainder of this paper is structured as follows: a section of related works (§2), followed by sections on description of the work (§3), evaluation (§4), future work (§5) and finally, the conclusion (§6).

II. Related works

Federated Learning, ever since its inception, has been a very popular topic of research. Work that we have come across on the same subtopic of Federated Learning (lack of heterogeneity) have proposed a variety of schemes to reduce the memory requirements of the Federated Learning Algorithm. Some papers try to tackle the problem by working with the averaging algorithm and the aggregation process [6, 8]. Other papers proposed different mechanisms to tackle the problem like reduction of uplink communication costs [7], and protocols using deadline-based approach to add efficiency to the system [9]. Some papers help the problem by reducing the requirements of the clients while they attempt to scale the model [5]. While we want to achieve these results as well, we try to only modify the model to produce its subset in an attempt to contribute towards the solution our initial problem.

A number of works that we have come across have also had a similar objective and methodology to ours. In fact, one of the most similar works to us [10] addresses the issue of lack of heterogeneity by introducing two novel strategies to reduce communication costs: the use of lossy compression on the global model sent server-to-client; and Federated Dropout, which allows users to efficiently train locally on smaller subsets of the global model and also provides a reduction in both client-to-server communication and local computation. Our objective and methodology overlap with Federated Dropout on many levels; however, Federated Dropout drops neurons in each layer. We reduce the model by reducing the total number of features sent to a client. Further, [11] proposes an efficient federated learning framework based on variational dropout. However, this paper deals with model sparsification which aims reduce the computational intensity of deep neural networks by pruning out redundant model parameters. We have not tackled this aspect of the model as of yet.

Other works have tried to reduce the processing required in the context of deep neural networks by introducing dropouts [12], sharing data movement and computation costs across networks [13] and freezing layers [14]. We borrow insights from these works and try to apply them to the Federated Learning scenario.

III. Description of the work

The logical way to begin this section would be by defining what we mean exactly by a subset of the model. Every model has an input layer and output layer whereas some consist of an intermediary hidden layer(s) also. Inputs to the input layer are the features. By subset of a model, what we mean is that the hidden and output layer stay the same, rather the number of features given to the input layer have been reduced. This way, the new subset model classifies into the same category as the original model. The only difference is that it uses a lower number of features to make this classification. The reduction in number of features reduce the computational workload significantly. If the number of features were to be reduced by half, then the number of parameters would reduce approximately by an order of 2 also, since the weights matrix gets reduced into half also. This can be seen in terms of the time for computation utilized by devices receiving the new model as compared to the time taken by those devices that received the complete model. By reducing the number of features by half, approximately 4 times reduction in time of the devices with the subset model is seen as shown in Fig. 1.

The foundation of our work lies in the classifying of the available devices into two categories: High-end and Low-end. High-end devices are devices with sufficiently enough resources and get the complete model parameters from the central server whereas low-end devices are resource constrained devices and get a subset of the parameters for processing and so they contribute towards the updating of only a fraction of the total parameters. Once a subset model is created and the classification of devices has been established, the Federated Learning framework can begin its progression.

The federated framework consists of a central server that coordinates the central, globalized model with multiple clients. At each outer iteration, a subset of the devices is selected by the server. This selection of devices consists of high-end as well as low-end devices. High-end devices receive the complete version of the model whereas for low-end devices, a random selection of a fixed number of features, lesser than the original number of total features, occurs that is used to produce a model dynamically at each round for each low-end device. The devices then carry out processing the model on their locally available data and once done, communicate their local model updates to the central server. The central server aggregates the updates after some initial preprocessing- the updates from low-end devices require some preprocessing before aggregation to match dimensions with the updates from high-end devices- and updates the global model accordingly.

IV. Evaluation

The evaluation of our work has been done using a simulation which calculates the number of high-end and low-end devices based on a percentage called the drop rate. The droprate is equal to the percentage of low-end devices. In our experimentation, we alter the drop rate and the number of features and analyze the effect on accuracy and loss as compared to the FedAvg framework. We also analyze the computational relaxation that low-end devices receive with the low-end subset model.

All experimentation has been done on a macOS operating system. The dataset used is non-IID and the total number of features in the dataset is sixty. The dataset used had no hidden layers.

A. Accuracy of Our Work versus FedAvg

We note that for lower drop rates (10% to 50%), at half number of features, our accuracy is comparable to that of the FedAvg framework. At higher percentages, our accuracy deteriorates significantly. We deduce that perhaps our proposed framework is not sufficient enough to contribute majorly to the model. Results can be seen in Figures 2 and 3.

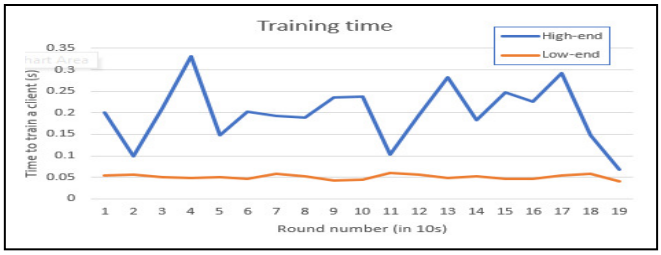

Fig.1.Processing Times of High-end and Low-end Devices

We notice a similar behavior when the number of features in the subset model reduce to ten. Higher number of features were not testes because the client computation was growing significantly higher which would defeat the overall purpose of the subset mode.

B. Loss of Our Work versus FedAvg

Results for losses follow the same trends as those in accuracy. However, one thing to note is that at lower percentages, for both ten and thirty number of features in the subset model, losses are more stable (less jagged lines in the figures) and are in fact a bit lower than the FedAvg's losses. Results can be seen in Figures 2 and 3.

C. Reduction in Computation for Low-end Devices

Low-end devices receive a smaller number of features- in essence, their input layer has a reduced number of neurons and so overall, there is a reduction in the model size as well as the number of parameters. To test the difference in computation required with the subset model as compared to the original model, we carried out a timed experiments where in every round, a randomly picked high-end device's and a randomly picked low-end device's computation time was recorded for two hundred rounds. The low-end device was given a model with half the number of features i.e. thirty. The results are shown in Fig. 1. By reducing the number of features by half, approximately four times reduction in time of the devices with the subset model is seen.

As a consequence of reducing the number of features to half, the network usage of low-end device has also been reduced since they now only need to communicate about half the number of parameters to the model. This is an extremely important property to achieve given the context of developing regions where internet connectivity is not only expensive but also unstable in most scenarios.

Results for half the number of features are similar to those of ten number of features in terms of timing. Network usage however is further lower for 10 number of features.

V. Future Work

Our work without doubt is a work in progress. We wish to shape our current work into a full-fledged framework that can function as a suitable alternative to the original FedAvg framework. To achieve this some of our future goals that we wish to add on our current work are discussed below.

a) Improve Subset Model As apparent from our evaluation, above a certain percentage drop rate (around 50%), our model fails to achieve the accuracy and convergence as the original FedAvg model does. We suspect that time might be due to the fact that perhaps the subset model is not sufficient to completely generate/update the centralized model. For this reason, we plan on improving the methodology of producing the subset model. Some ideas we have in mind are further randomization of the features- perhaps in the shape of a new feature selection protocol that might assist and automate the randomization process. We also would like to try out pruning of parameters within the subset model. For this, we would first have to analyze which parameters are more significant and which are comparatively not so significant and so can be pruned.

b) Freezing Layers Freezing layers have shown remarkable results within the context of neural networks. Thus, we would like to mold the concept and merge it in the context of federated learning, especially for models given to the low-end devices. We are fairly optimistic about this adding in positive improvements to our work.

c) Binary Weights We would like to explore the effects of creating a third category of devices that is reserved for devices that have very poor specs. For this new class, we would like to try out the effect of processing weights in a binary manner i.e. a parameter could be either take the value 0 or 1. We are unsure of what to expect but hope to see meaningful results.

VI. Conclusion

This paper introduced a new technique to attempt and overcome the key problem of the lack of high degrees of systems and statistical heterogeneity in the state-of-the-art FedAvg- it aims to make federated learning possible in developing regions. Our work can be looked at as a modification of the FedAvg, the current state-of-the-art method for federated learning. The methodology that the paper proposes is that clients be divided into two categories based on the availability of resources. The devices with plenty of resources receive a complete model to process whereas the other category of devices receive a subset of the model with a fewer number of features. Our results show that this setup can achieve comparable accuracies and convergence times with the original federated learning framework, whilst reducing the workload on resource constrained devices significantly. We also note a slight stability and decrease in losses with our proposed framework. We further note that reducing the number of features from half to one sixth does not have any significant effects on our results.

Acknowledgment

We would like to thank our instructors Ihsan Ayyub Qazi, Zafar Ayyub Qazi and Zartash Afzal Uzmi for their tremendous efforts towards their students.

References

[1] Mcmahan et. al, "Communication-efficient learning of deep networks from decentralized data," *AISTATS*, 2017.

[2] White House Report. "Consumer data privacy in a net- worked world: A framework for protecting privacy and promoting innovation in the global digital economy," *Journal of Privacy and Confidentiality*, 2013.

[3] "Measuring digital development: Facts and figures 2019," https://www.itu. int/en/ITU-D/Statistics/Pages/facts/default.aspx., 2019.

[4] Chris DeBrusk, "The risk of machine learning bias (and how to prevent it)." https://sloanreview.mit.edu/article/the-risk-of-machine-learning-bias-and-how-to-prevent-it/, 2018.

[5] Bonawitz et. al, "Towards Federated Learning at Scale: System Design," *arXiv:1902.01046,* 2019.

[6] Li, Tian & Sahu, Anit & Zaheer, Manzil & Sanjabi, Maziar & Talwalkar, Ameet & Smith, "Federated optimization for heterogeneous networks," *Virginia,* 2019.

[7] Mcmahan et. al, "Federated learning: strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[8] Wang, Shiqiang et al., "Adaptive federated Learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications. PP. 1-1. 10.1109/JSAC.2019.2904348*, 2019.

[9] Takayuki Nishio and Ryo Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," *arXiv preprint arXiv:1804.08333*, 2018.

[10] Sebastian Caldas, Jakub Konecˇny, H. Brendan McMahan, Ameet Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *arXiv:1812.07210v2*, 2019.

[11] Unknown, “Efficient federated learning via variational dropout,” unpublished.

[12] Nitish Srivastava et al., “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research,* 2014.

[13] Abdul Wasay, Brian Hentschel, Yuze Liao, Sanyuan Chen, Stratos Idreos, “Mothernets: rapid deep ensemble learning,” *arXiv:1809.04270v2* , 2020.

[14] Brock, A, Lim, T, Ritchie, JM & Weston, “Freezeout: accelerate training by progressively freezing layers,” *N.J.,* 2017.

A. **Graphs and Figures**

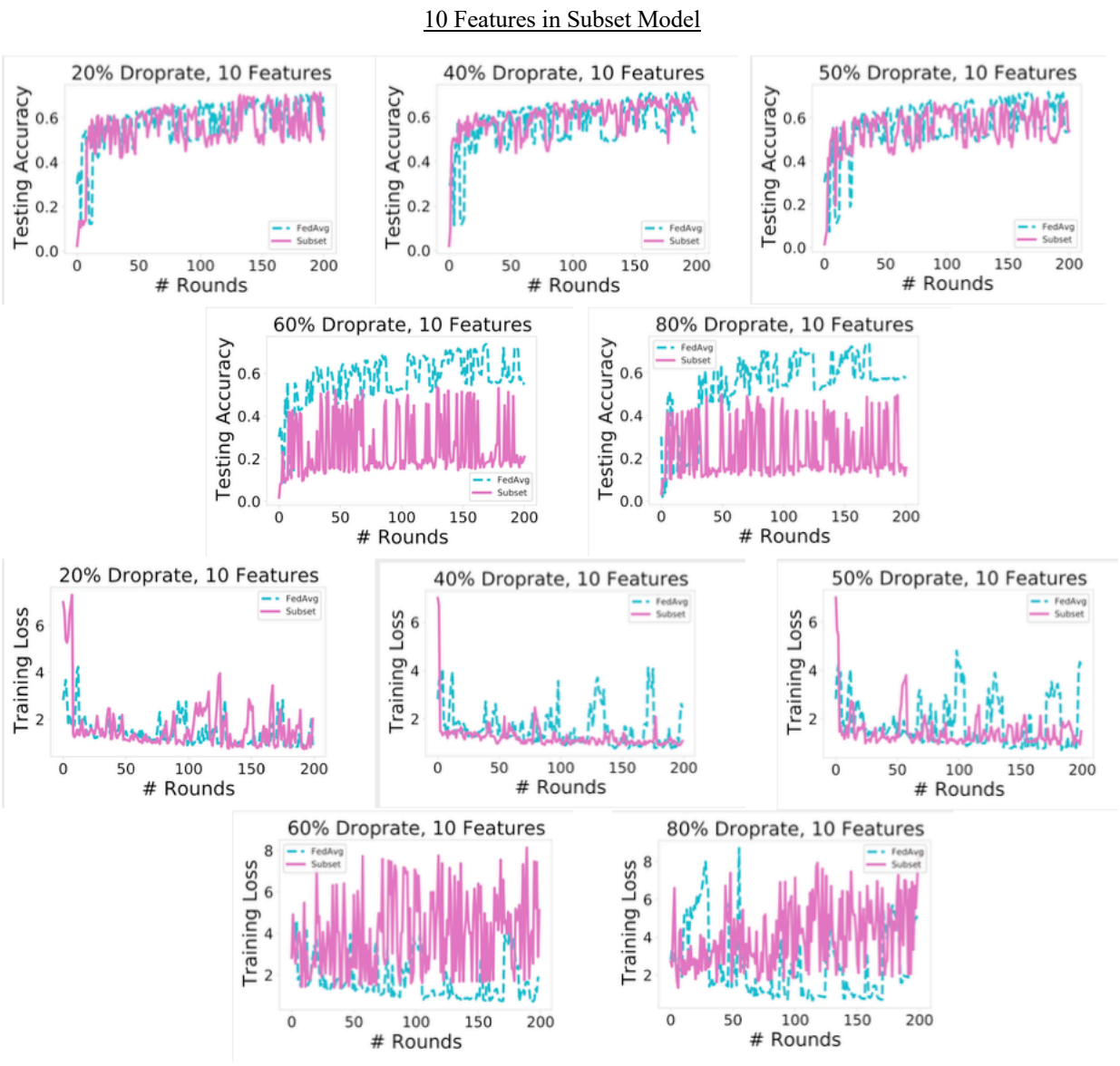

Fig.2. Accuracy and Loss Graphs when the Subset Model had 10 Features only.

30 Features in Subset Model

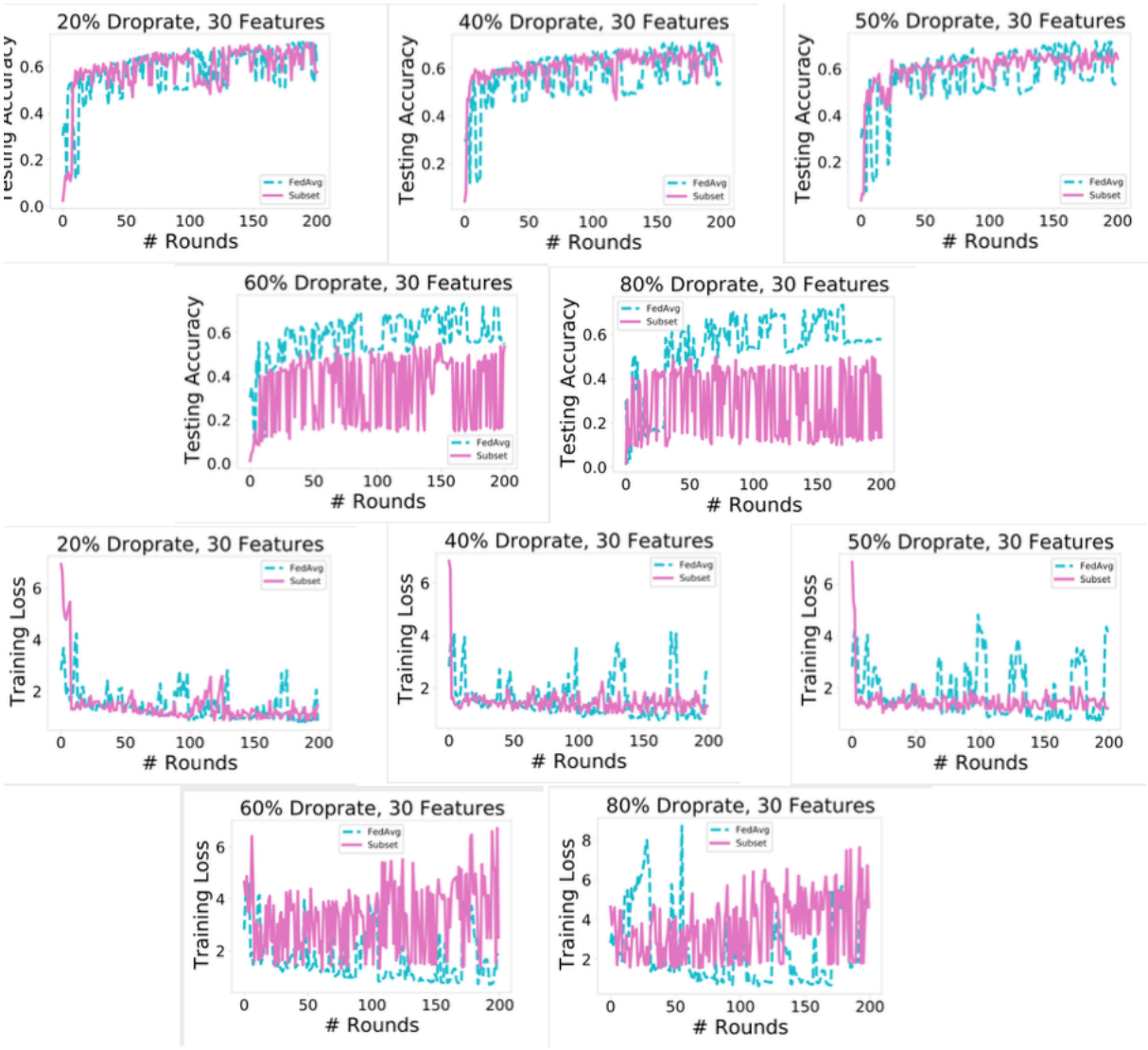

Fig.3. Accuracy and Loss Graphs when the Subset Model had 30 Features only.