Research Proposal

Mohammad Maaz Owais, Muhammad Hamza Khawaja, Muhammad Taha, Shazer Ali

1. Introduction

Federated Learning involves training a shared global model using local data and compute on various user devices. Several approaches have been proposed to implement this paradigm starting with Federated Averaging [2]. However, not many of these approaches consider low-end devices that are unable to perform training. This has implications for fairness due to introduction of systematic bias, in addition to degradation in model accuracy. Recent works such as FedProx (2) and Hassas (3) have attempted to include slow devices by incorporating partial work and serving a subset model according to device characteristics, respectively. These approaches have been evaluated on large-scale simulations using LEAF Benchmark [1] and a small-scale testbed of mobile devices in case of Hassas. To the best of our knowledge, none of these works have been evaluated on mid-scale federated learning systems using actual mobile devices with a sufficiently large number of clients.
Significance of the problem stems from the need to include low-end smart phones and IOT devices in the process of training the data. Inclusitivity of such users will allow for a more precise model to be generated as the training data will be diverse and more reflective of real life. Moreover, in the developing countries where more than 57% of population are categorised as low-end users, their exclusion in the past models [3] will lead to great inaccuracies and biasness. The approach of having sub-models in these devices will increase the likelihood for to train models even in the case of limited poor connectivity and limited bandwidth. Overall, the concept of Hasaas will create a better privacy preserving and resource efficient model.

2. Related Work

In recent times, there have been several studies done on the algorithms and frameworks to study federated learning. **FedProx** is an optimized version of the Federated Average algorithm that is more robust than the original, especially in case of highly heterogenous settings. **FedAdaptive** presents federated versions of adaptive optimizers such as Adam and Adagrad. These advances in the FL framework have mostly leveraged the **LEAF** simulation framework. Few works have performed evaluations on real-world devices. A recent paper in 2022 **"An Experiment Study on Federated Learning Testbed"** performance of federated learning on IOT devices is an example in that direction. Similarly, the paper **"Optimize Scheduling of Federated Learning on Battery-powered Mobile Devices"** utilized a testbed of 4 devices with varying CPU cores and processing capabilities to to evaluate their approach using the MNIST dataset. To facilitate experiments on real devices, **Flower** a highly developed framework has been introduced that can support millions of real world clients using just a few high end GPUs.

3. Design

To our knowledge, our work is the first large-scale FL deployment on real-world user devices that will aid evaluations for academic purposes.

3.1. Application Layer

1. Frontend
2. Data Cache
3. Prediction Model

Our application in target will be a specific University dedicated chatbot which handles a user's questions related to Graduate and Undergraduate level applications and academic assistance. Essentially our user will interact with our chatbot over a mobile application, where question will be provided as input. The question will be given as an input to our Deep Neural Network model on the user's device which will then produce an output as answer [4]. We will require user to provides us with the feedback for the query related answer, to improve the model. Our application will perform some preprocessing and then cache each question and answer pair with the feedback given, to later allow for the model to be trained on the device.

3.2. FL Platform Layer

1. **FL Client Runtime**
   - Model Training
   - Device Analytics
     No personally identifiable information will be logged.
     - Logs
     - Device OS
     - Device Model
     - Device State
     - Training time
     - Memory profile
     - Battery profile
2. **FL Server Runtime**

3.3. Coordination Layer

3.3.1. Device Management :
The selection and reporting of devices for FL tasks are part of device management. In order to make the device management process inclusive, we add functionality to the mechanism described in the paper Towards Federated Learning At Scale: System Design. Devices that meet the

eligibility requirements check in with the server for any FL tasks that are open as part of the selection process. After allocating the FL tasks to available devices, the server waits for participants to report updates. To signal the end of a round, the server uses a goal counter and timer. The goal count is split into two quorums, one for high-end devices and the other for low-end devices, both of equal size. The round is marked complete if the goal count is achieved within the timeout value else the round is discarded.

3.3.2. Failure Detection :

The device is declared failed if it violates the eligibility criteria during a FL round (unmetered connection, plugged in to power). During the FL round, the failure is detected using the ping and ack mechanism between the server and the device. This aids the server in keeping track of active participants of a FL round.

4. Methods

The tasks we need to accomplish can be broadly categorized as follows.

1. Application Development
2. FL Platform Development
3. User Base Development

4.1. Phase I

We will begin by developing an android application. Simultaneously, we will develop an end-to-end prototype of the FL platform for 4 devices. Once a working prototype is in place, the next step will be to scale it to 20 users and perform thorough end-to-end testing. The primary objective is to ensure efficient, seamless deployment of the FL platform on a sufficiently large number of clients.

4.2. Phase II

Once this is achieved, we will integrate our android application with the FL platform.

4.3. Phase III

Finally, we will make an effort to scale our application to 100+ users.

5. Timeline and Work Division

Timeline and division of work.

References

[1] CALDAS, S., WU, P., LI, T., KONEČNÝ, J., MCMAHAN, H. B., SMITH, V., AND TALWALKAR, A. LEAF: A benchmark for federated settings. *CoRR abs/1812.01097* (2018).

[2] MCMAHAN, H. B., MOORE, E., RAMAGE, D., AND Y ARCAS, B. A. Federated learning of deep networks using model averaging. *CoRR abs/1602.05629* (2016).

[3] NASEER, U., BENSON, T. A., AND NETRAVALI, R. Webmedic: Disentangling the memory-functionality tension for the next billion mobile web users. In *Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications* (New York, NY, USA, 2021), HotMobile '21, Association for Computing Machinery, p. 71–77.

[4] VAMSI, G. K., RASOOL, A., AND HAJELA, G. Chatbot: A deep neural network based human to machine conversation model. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (2020), pp. 1–7.