# CIS520 Project Proposal

## A validation of decentralized federated learning for clinical computer vision

Jordan Matelsky, Felipe Parodi
Team Name: Fed Heads

## 1 Motivation

Among the most inspiring of recent advances in computer vision (CV) is the extraordinary performance of deep neural networks on clinical radiology imagery: neural networks can now perform radiologist-level disease classification from imagery alone [1]. One major setback in the field of clinical CV is the need for diverse and extensive training data. Clinical datasets are protected under local and federal privacy law, which dramatically increases the difficulty of amassing a large, multi-site training dataset for deep learning researchers.

Federated learning (FL) is a technique in which multiple compute nodes independently train the same neural network architecture, and then "agree" upon network parameters through a centralized fusion step [2]. In recent work, this process has been adapted to a *decentralized* (DFL) approach in which each compute node performs its own local fusion step (rather than relying upon a central authority). [3] We propose an application of this novel DFL approach to the domain of clinical CV. We intend to demonstrate that DFL learning on distributed datasets — *i.e.*, cooperatively learning a shared model across hospital boundaries — is a viable replacement for the much more logistically and legally nuanced aggregation of data in a centralized repository. We will compare DFL approaches with the traditional centralized approach, and determine what performance penalties — if any — are imposed by the novel training architecture. Upon completion of this work, we intend to develop a go/no-go suggestion for the use of DFL for clinical research based upon the evaluation metrics detailed in *Evaluation*, below.

## 2 Data

In order to compare decentralized federated learning and traditional ML approaches, we will use an existing, well-studied clinical CV dataset. Ideal options include the **NIH Chest X-ray** dataset [4] or **CheXpert** [1]. In the case that these large datasets and their reference network complexities overextend our compute resources, we have identified several smaller binary classification datasets, such as the **Pneumonia Chest X-Ray Images** dataset on Kaggle [5], as viable alternatives, in order to de-risk our project. A summary of these datasets is available in **Table 1**.

## 3 Related Work

Centralized federated learning (FL) has been in active production use for many years. Perhaps the most commonly encountered implementation is *GBoard*, a virtual keyboard that trains a local predictive network locally on each phone, the parameters of which are then uploaded to a central Google server to be fused and re-distributed. [2] Decentralized federated learning (DFL) is a novel approach that removes the central server and instead enforces a peer-to-peer network topology on the federated compute nodes. Though this application has been explored experimentally in the past, it has not to our knowledge been applied in practice, nor in a production environment. [6, 7]. The first known generalized framework for DFL was released in 2021, which we intend to use here [3].

## 4 Problem Formulation

We will examine two types of machine learning in this work: Traditional learning, and decentralized federated learning (**a** and **c** in **Fig. 1**). We will compare the performance of a reference network (the same between each type of learning) by modifying its access to data in order to simulate real-world data imbalances.

| Dataset | Dataset Size | Task Size | Reference |
|---|---|---|---|
| NIH Chest X-ray | 108,948 Images | 8 Multilabels | [4] |
| CheXpert | 224,316 Images | 14 Multilabels | [1] |
| Chest X-Ray Images (Pneumonia) – Kaggle | 5,863 Images | Binary Classification | [5] |

Table 1: Options for dataset use. Two larger datasets are viable candidates for multilabel assignment; a binary classification example dataset is also provided as a simpler alternative. All options have several high-performing reference network implementations available online.

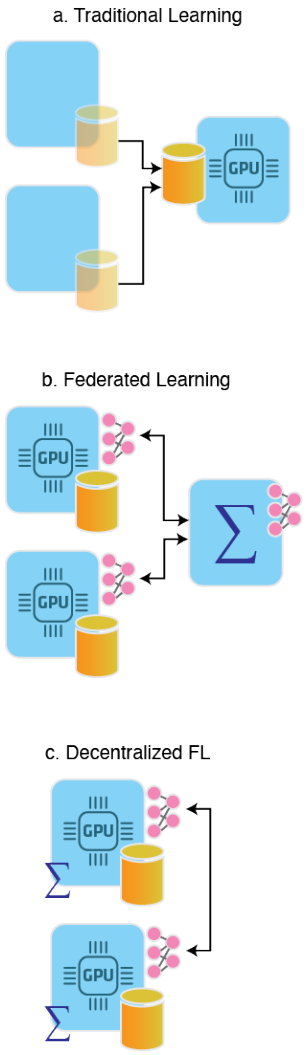


Figure 1: Three types of learning examined in this work. **a. Traditional learning.** Edge nodes provide only data. A central server performs all computation. **b. Federated learning.** Edge nodes perform local computation and training. A central server performs parameter fusion. **c. Decentralized federated learning.** Edge nodes perform local computation and training. Each node performs fusion with its networked neighbors.

We will first separate the data into multiple hospital "sites," where each hospital has a different balance of each class label. (For example, the Hospital of the University of Pennsylvania sees more *pneumonia* patients and no *edema* patients; the Johns Hopkins hospital sees *edema* patients but no *pneumonia* patients; etc.) We will then provide the network access to data thus:

**Traditional Learning.** We will provide the network serial access to all of the data, randomly shuffled (the control case, to illustrate the conventional approach to learning on large datasets). Then we will provide a new, untrained network with *serial* access to each of the split hospital datasets, one after the other, to illustrate the closest analog of federation on a single node.

**Decentralized Federated Learning.** We will provide each node its own dataset from the split detailed above. The nodes will communicate in an all-to-all network topology for peer-to-peer parameter fusion.

Following training, we will compare the performance of each network in order to determine which of the methods best captured the variance of the training data, and we will then make suggestions for clinical application of federated learning based upon the strengths and weaknesses we discover.

In our analysis, we will also consider the implications of data transfer as well as differential privacy [8], two aspects in which federated learning has considerable strength.

# 5 Methods

We will first identify a target dataset of interest (see **Data**). We will then retrieve and validate the performance of a reference neural network model. Reusing a community-contributed model enables us to better contextualize against existing baseline performance, as well as de-risk the technical components of this work.

We will train each network architecture entirely independently, though we will reuse the same train/test data splits for each (see **Problem Formulation**).

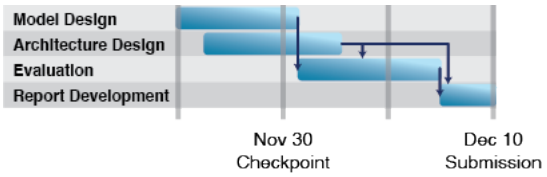


Figure 2: A Gantt chart of proposed project progress. More details are provided in **Project Plan**.

This will help to ensure that we are fairly comparing each network topology and training architecture.

Finally, we will compare the performance of each architecture, based upon the metrics detailed in **Evaluation**. We will report these, alongside a "go/no-go" analysis of the different networks.

# 6 Evaluation

We will evaluate the training architectures along several axes, to be determined in parallel with experimental design. In particular, we will measure loss curves over time, as well as common metrics such as precision, recall, and $F_1$. In addition to these conventional metrics, we will also plan to measure the volume of data transfer per model (i.e., the cost of bandwidth), the wall clock time to train, and the maximum achieved performance.

In addition to these quantitative metrics, we will also report on qualitative discoveries during the implementation of this novel approach. If FL tools are difficult to adapt on our intended timeline, this is a relevant and noteworthy weakness of the methods proposed, and we will note it accordingly in the evaluation.

# 7 Project Plan

We intend to adhere to the following project plan timeline (**Fig. 2**). Our ability to meet these deadlines with each of the techniques is a relevant and important criterion, and so technical engineering delays will be used as an additional, qualitative evaluation metric.

**November 30: Checkpoint.** At the checkpoint, we will have selected a dataset and benchmark model, and performed an initial evaluation of our selected model. We will also have network topologies designed for the federated learning use cases. After the checkpoint, we will finalize training of the FL architectures and begin development of the final report and notebooks.

**December 10: Final Submission.** At the time of final submission, we will present a reference notebook detailing our experimental findings. We will also provide a written manuscript with the intention of submitting for peer review following the conclusion of the class.

# References

[1] P. Rajpurkar, J. Irvin *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," 2017.

[2] B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," https://ai.googleblog.com/2017/04/federated-learning-collaborative.html, 2017, accessed: 2020-09-24.

[3] M. Wilt, J. K. Matelsky, and A. S. Gearhart, "Scatterbrained: A flexible and expandable pattern for decentralized machine learning," *In submission*, 2021.

[4] X. Wang, Y. Peng *et al.*, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[5] D. S. Kermany, M. Goldbaum *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[6] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," *arXiv preprint arXiv:1901.11173*, 2019.

[7] A. G. Roy, S. Siddiqui *et al.*, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *arXiv preprint arXiv:1905.06731*, 2019.

[8] M. Naseri, J. Hayes, and E. D. Cristofaro, "Local and central differential privacy for robustness and privacy in federated learning," 2021.