# A Federated Learning Framework for Healthcare IoT devices

Binhang Yuan, Song Ge, and Wenhui Xing

## [ Motivation ]

- The Internet of Things (IoT) revolution has shown potential to give rise to many medical applications.
- IoT has become one of the most important data source for many medical applications.
- IoT supports systems capable of continuously clinical-level monitoring of subjects' conditions and acquiring a variety of bio-signals.
- However, data owners are increasingly privacy sensitive.

## [ Challenges ]

- Traditionally, data collected by IoT devices are uploaded to a data center and further leveraged to train machine learning models.
- This violates privacy requirement regarding individually identifiable health information.
- Federated learning attempts to resolve this data dilemma.
- However, vanilla federated learning is not sufficient to meet the requirement of healthcare IoT devices with:
  - Limited energy storage;
  - Low computational capacity;
  - Restricted network bandwidth.

## [ Decomposed Federated Learning ]

- **Multiple Layer Neural Network**: A deep neural network is designed to approximate a target function $\mathbf{y} = f^*(\mathbf{x})$, which maps an input feature $\mathbf{x}$ to output prediction $\mathbf{y}$. Formally, the function $f^*$is composed by a chain of N different functions as:

$$f^*(\mathbf{x}) = f^N\left(f^{N-1}\dots\left(f^2\left(f^1(\mathbf{x})\right)\right)\right).$$

- **Vanilla Federated Learning**: A classic federated learning systems includes M data owners who need to train models $\{f_1, f_2, \dots, f_M\}$ on their datasets $\{D_1, D_2, \dots, D_M\}$, the aim is to minimize $f(x)$ w.r.t., parameter $w$:

$$\min_x f(w) = \sum_{j=1}^{M} f_j\left(x \mid D_j\right).$$

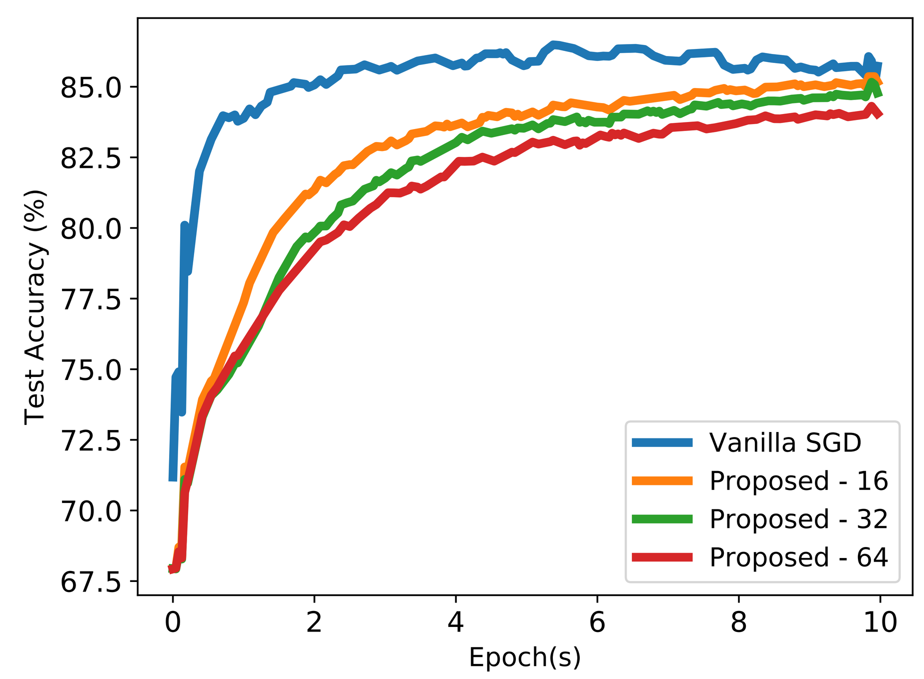
- **Decompose the Neural Network:** the approximated function $f^*$ is decomposed so that each IoT device (indexed by j) will include a local version of the first shallow component $\mathbf{a}^1 = f_j^1(\mathbf{x})$, while the rest part $\mathbf{y} = f^N\left(\dots\left(f^2\left(\mathbf{a}^1\right)\right)\right)$ is allocated on the centralized server:

$$\min_w f(x) = \sum_{j=1}^{M} f^N\left(\dots\left(f^2\left(w^{2,\dots,N} \mid f_j^1\left(w_j^1 \mid D_j\right)\right)\right)\right)$$

- **Sparsify Activations and Gradients**: To further reduce the network traffic, we extend the idea of sparsification of gradients. we sparsify $\mathbf{a}^1$, $d\mathbf{a}^1$ by only communicating the top K ( $\leq 10\%$) elements at each iteration.

## [ Preliminary Evaluation ]

- Benchmark problem:
  ResNet34[1] for PhysioNet 2017.
- Convergence of ResNet in PhysioNet 2017:




- Network traffic comparison for each iteration:

| | FedAvg [2] | SplitNN [3] | Proposed |
|---|---|---|---|
| 16 Device | 1.36GB | 32MB | 3.2MB |
| 32 Device | 2.72GB | 64MB | 6.4MB |
| 64 Device | 5.45GB | 128MB | 12.8MB |

## [ References ]

1. Bourn, C., Turakhia, M. P., and Ng, A. Y. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. Nature medicine, 25(1):65, 2019.
2. Konecny, J., McMahan, H. B., Ramage, D., and Richtarik, P.Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527, 2016.
3. Vepakomma, P., Gupta, O., Swedish, T., and Raskar, R. Split learning for health: Distributed deep learning without sharing raw patient data. arXiv preprintarXiv:1812.00564, 2018.