Benchmarking and Optimizing Federated Learning with Hardware-related Metrics

Kai Pan[1], Yapeng Tian[2], Yinhe Han[1], Yiming Gan[†1]

[1] Institute of Computing Technology Chinese Academy of Science, Beijing, China
[2] The University of Texas at Dallas, Texas, USA

[†] Corresponding Author

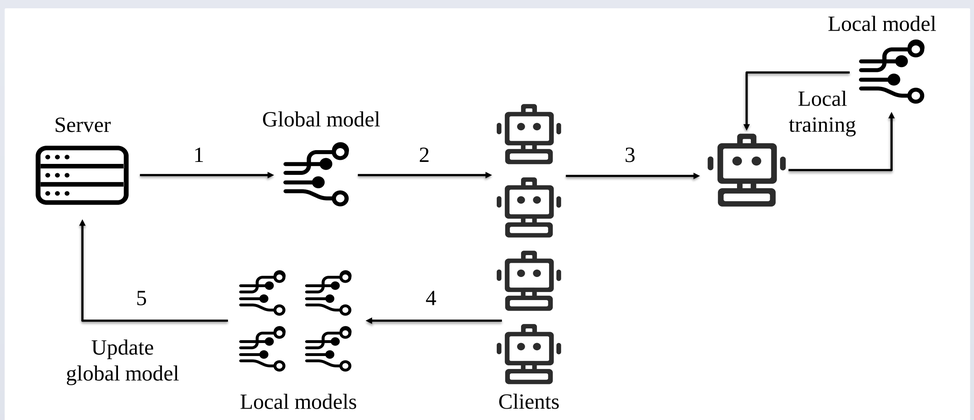
ABSTRACT

Federated learning (FL) serves as an effective way of preserving data privacy at net work training through offloading training tasks to different client hardware and aggregation. Real hardware-related metrics such as latency and energy consumption directly decide the performance and accuracy trade-off in federated learning frameworks, yet most FL optimizations do not use real hardware metrics.

In this work, we propose to benchmark federated learning with real measured hardware metrics and optimize FL frameworks through tailoring training hyper-parameters before offloading tasks each round given hardware metrics. With two examples FedAvg and FedOpt, we demonstrate we can significantly save training energy by up to 97.2% and training latency by up to 96.0% while maintaining training accuracy. Source code can be found at https://github.com/RLC-Lab/FEDHW.git.

Contributions:

- We systematically propose an end-to-end characterization of federated learning algorithms with real hardware metrics such as training latency and energy consumption. We break down the latency and energy consumption of individual components and the results show that a significant amount of latency and energy are spent on the training process happening on local clients and central server processing.
- We propose a framework that collects data from both the server to maintain accuracy and real hardware to reduce latency and energy consumed in the federated learning process. The scheduling process is formulated as a constraint optimization problem.
- We implement two instances of our optimization framework with a simulated annealing optimizer and a genetic algorithm optimizer. We significantly improve the total latency and energy consumption of federated learning without losing any accuracy.



BENCHMARKING

A. Energy Modeling

a) Energy consumption for server computation: Servers typically employ high-end CPUs and GPUs for the operations. The total energy consumption on the server is typically the sum of energy from CPUs and GPUs, denoted as $E_{server} = E_s^{cpu} + E_s^{gpu}$.

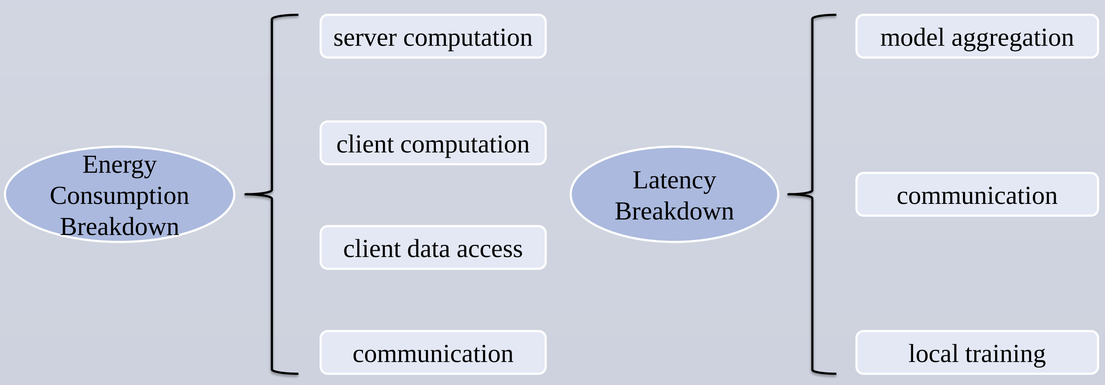
b) Energy consumption for client computation: The computations are mainly matrix multiplication and accumulation (MAC). Energy consumption for client computation can be either consumption measured using tools like power management firmware or calculated with a cycle-accurate hardware simulator such as SCALE Sim v2.

c) Energy consumption for data access: The most common memory hierarchy is usually composed of a Dynamic Random Access Memory (DRAM), a cache, and a Register File (RF). In neural network training, inputs, weights, and activation maps are loaded from DRAM to cache, and from cache to RF. Thus, the energy consumption for data access are sum of three components E_{DRAM}, E_{Cache}, and E_{RF}. In reality, E_{Cache} and E_{RF} are usually taken into consideration when measuring the energy consumption of the accelerators. E_{DRAM} needs to be measured separately in most cases.

d) Energy for Communication: Assuming clients and servers are communicating through an ideal Wi-Fi protocol, the communication energy can be calculated using $E_{comm} = P_{wifi} * \frac{DataSize}{BitRate}$, where P_{wifi} stands for the power of the Wi-Fi module in the client, DataSize stands for the amount of data sent to the server, and BitRate stands for transmission speed of the Wi-Fi module.

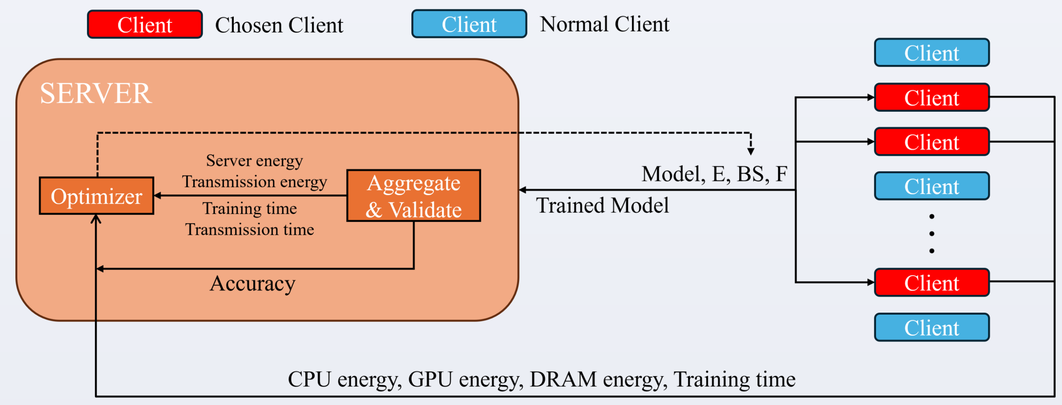
B. Latency Modeling

The federated learning training process needs to undergo T times of "local training – communication – model aggregation" process. For each round, the latency L is determined by the slowest client, which is calculated using $L = L_{server} + L_{client} + L_{comm}$. The entire training latency is formulated by $T \times L$.



OPTIMIZATION FRAMEWORK -- FEDHW

FEDHW seamlessly integrates with any federated learning framework, operating on the server side. It continuously gathers inputs like latency, energy consumption, and model accuracy, and then uses this data to generate scheduling policies for the next round of training.



FEDHW optimization framework

We propose two optimizers: FEDHW-SA, which formulates the optimization process as a simulated annealing (SA) algorithm, and FEDHW-GA, which employs a genetic algorithm (GA) approach.

EXPERIMENT & EVALUATION

Dataset	Model	Parameters	FedAvg_Acc	FedOpt_Acc
MNIST	CNN	(E=20, BS=10, F=0.1, NC=100)	98%	98%
CIFAR10	ResNet-18	(E=5, BS=64, F=0.1, NC=100)	90%	90%
ESC50	M18	(E=20, BS=10, F=0.2, NC=50)	68%	68%
R8	LSTM	(E=50, BS=5, F=0.2, NC=50)	92%	92%

We train all four models using their default settings with both FedAvg and FedOpt. Each client's dataset is Independent and Identically Distributed (IID). We utilize NVIDIA Jetson TX2 embedded devices as the client platform, featuring a 256-coure NVIDIA Pascal GPU architecture. For the server, we employ a desktop equipped with an Intel i9-12900K CPU and an NVIDIA RTX 3090 GPU. Communication between the server and clients is established through Wi-Fi.

As shown in the tables below, the energy consumption can be saved up to 97.2%. Compared to FedAvg, FEDHW reduces energy consumption by 53.5% by tailoring E, BS, and F in each round, while for FedOpt, the reduction is 59.2%. Specifically, FEDHW-SA achieves a 60.6% reduction in energy consumption, while FEDHW-GA achieves a 59.2% reduction. We observe that FEDHW-GA outperforms FEDHW-SA on larger datasets, as the GA optimizer tends to converge to optimal settings more quickly. Conversely, FEDHW-SA performs better on smaller datasets, as its optimization process results in more aggressive hyperparameters which means smaller E and BS, leading to faster model aggregation and lower energy consumption.

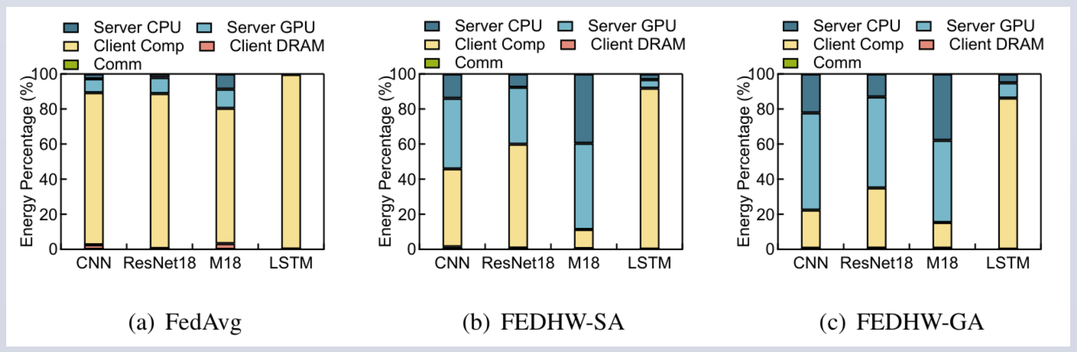
The training latency can be reduced up to 96.0%. On average, FEDHW reduces training latency by 79.3% compared to the FedAvg baseline and 78.0% compared to FedOpt.

Model	FedAvg	SA	GA	FedOpt	SA	GA
CNN	1x	0.43x	0.98x	1x	0.23x	0.29x
ResNet-18	1x	0.34x	0.22x	1x	0.61x	0.63x
M18	1x	0.69x	0.90x	1x	0.62x	0.52x
LSTM	1x	0.08x	0.07x	1x	0.08x	0.03x

Energy Consumption Optimization

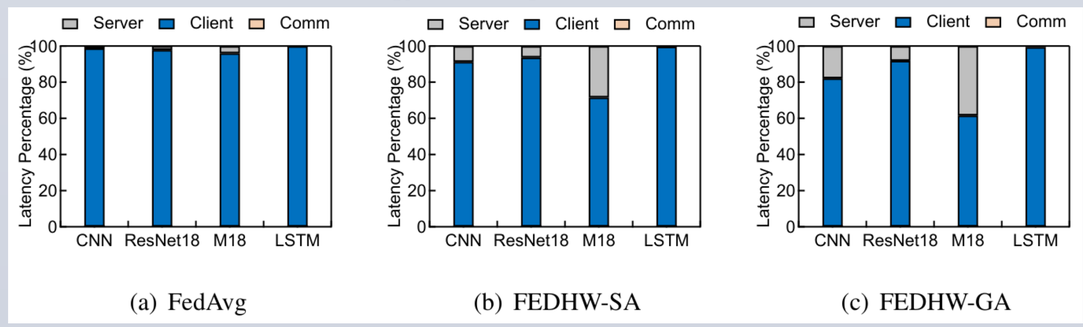
Model	FedAvg	SA	GA	FedOpt	SA	GA
CNN	1x	0.25x	0.64x	1x	0.23x	0.29x
ResNet-18	1x	0.34x	0.28x	1x	0.70x	0.78x
M18	1x	0.79x	0.61x	1x	0.46x	0.15x
LSTM	1x	0.21x	0.17x	1x	0.05x	0.04x

Latency Optimization



(a) FedAvg (b) FEDHW-SA (c) FEDHW-GA

Energy Consumption Breakdown



(a) FedAvg (b) FEDHW-SA (c) FEDHW-GA

Latency Consumption Breakdown

ACKNOWLEDGEMENT & CONTACT

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62025404).

Yiming Gan, PhD, Assistant Researcher, ganyiming@ict.ac.cn