Hyperdimensional Computing on IoT Devices

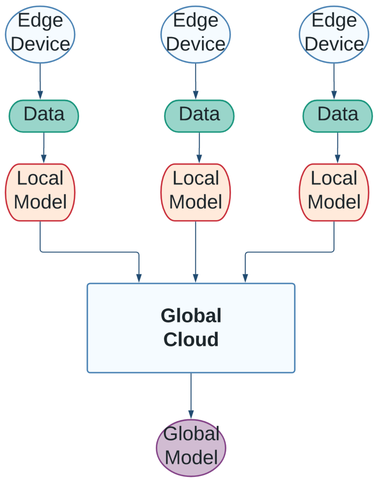
UC San Diego
JACOBS SCHOOL OF ENGINEERING
Computer Science and Engineering

Quanling Zhao, Kai Lee, Jeffrey Liu, Muhammad Huzaifa
Advisors: Xiaofan Yu, Tajana Rosing

ERSP
UC San Diego
Early Research Scholars Program

BACKGROUND & MOTIVATION

- Federated Learning is a distributed machine learning paradigm that is most used where data is distributed across many devices such as cell phones, or multiple computer networks.



Local data from the edge devices are used to generate local models.

These models are then sent to the global server.

The server aggregates the models to generate the global model.

- Current federated learning paradigms are bulky and costly, so it is difficult to implement such systems onto low power devices that have limited resources and power.
- One avenue to reducing the cost of FL is Hyperdimensional computing (HDC), a lightweight and resource efficient paradigm.
- HDC encodes data into high dimensional vectors called **hypervectors**. Training is performed by comparing query vectors with class vectors.

Animal Classification with HDC

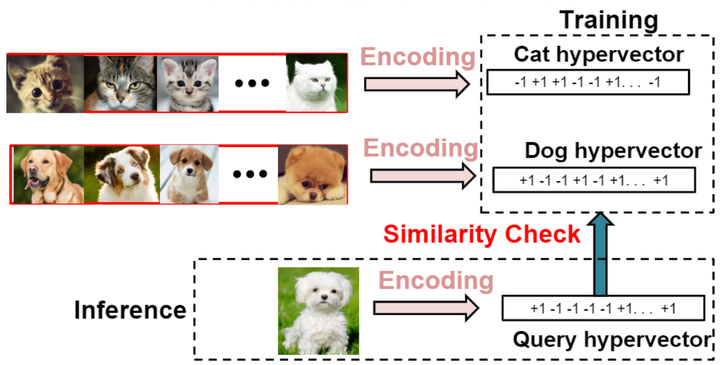


Figure courtesy of UCSD SEElab

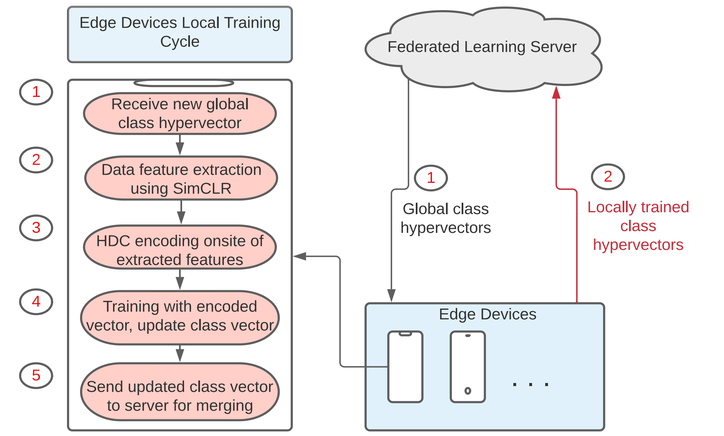
- Datasets are encoded to class hypervectors.
- Query images are encoded and categorized to the most similar class hypervector in one-pass training.

PROBLEM STATEMENT

We aim to improve federated learning systems with the addition of HDC in place of a neural network such that these systems may perform better on low power devices. We will evaluate our work in terms of **accuracy, communication cost and energy efficiency, and robustness against noisy communication channels.**

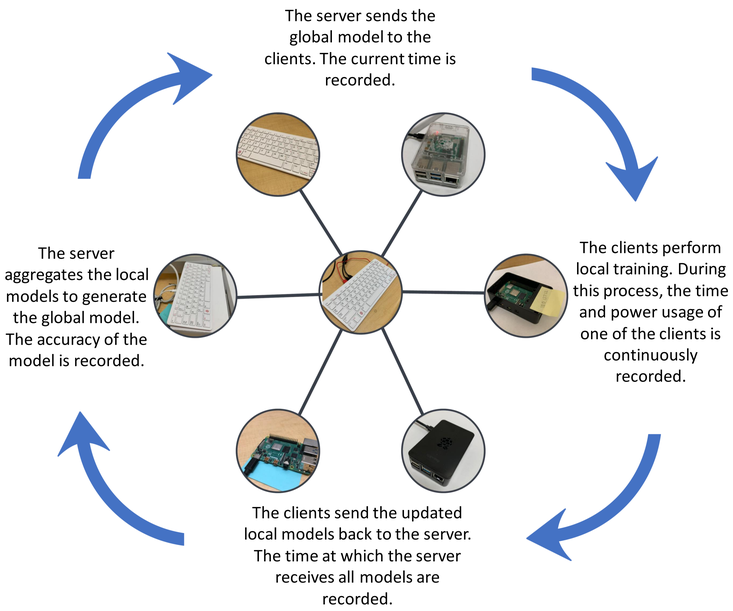
We used FedML, which is an existing Federated Learning framework, in a real deployment of Raspberry Pi's. We then modified FedML to have edge devices locally train hypervectors and send them to the server for aggregation after completing training.

Federated Learning System with HDC



Federated Learning with HDC Deployment

The new system was deployed on a network of **6** total Raspberry Pi 4 and Raspberry Pi 400 clients and **1** Raspberry Pi 4 server for testing and evaluation.

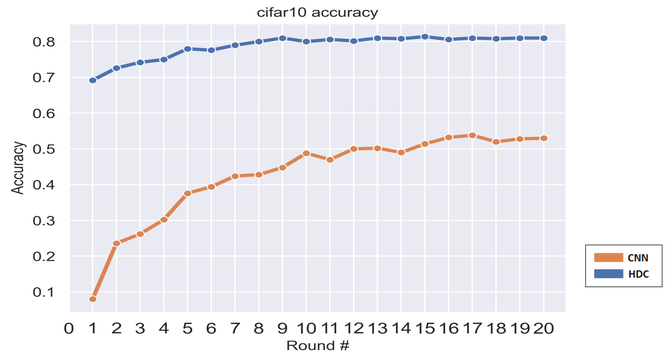


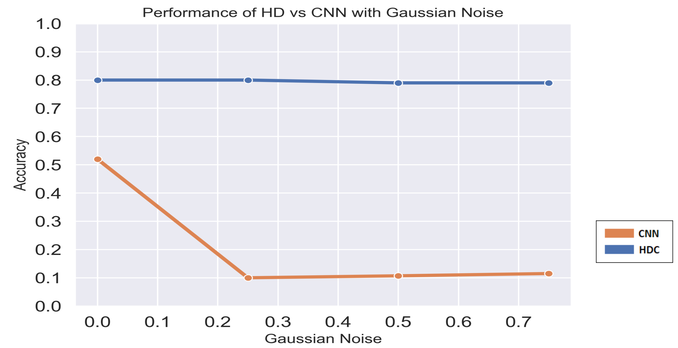
RESULTS AND EVALUATION

We measured the accuracy, power consumption, and model performance under different level of communication noisy (Gaussian Noise). In addition, we also measure the average communication size and time-to-converge for each model.

- For HDC, we used 1 local epoch whereas CNN used 10 local epoch.
- For both HDC and CNN, we used 6 client, 20 communication rounds and 500 local sample per client.

Accuracy of HDC vs CNN Measured Over Time





	Baseline Training Time (s)	HD Training Time (s)	Baseline Energy Usage (J)	HD Energy Usage (J)
CIFAR10	7117	3175	42296	19164

*Time: total time used for accuracy to converge.

Average Model Size for CIFAR10

CNN	12.2 Megabyte	HDC	1.9 Megabyte

Analysis

FL with HDC provide:

- Better Accuracy compares to Baseline on complex image dataset
- 6 times less communication cost
- Robustness in noisy communication channels

ACKNOWLEDGEMENTS

We would like to thank Christine Alvarado and Mai Elshereif, for providing the opportunity for us to participate in the Early Research Scholars Program.
We would also like to thank Xiaofan Yu and Tajana Rosing, for their guidance and mentorship throughout the course of this project.
We would also like to thank Rishikanth Chandrasekaran and Kazim Ergun, for sharing their research and making the deployment of the code to be possible.