ECAvg: An Edge-Cloud Collaborative Learning Approach using Averaged Weights

Atah Nuh Mih*, Hung Cao*, Asfia Kawnine*, Monica Wachowicz‡

*Analytics Everywhere Lab, University of New Brunswick, Canada

‡RMIT University, Australia

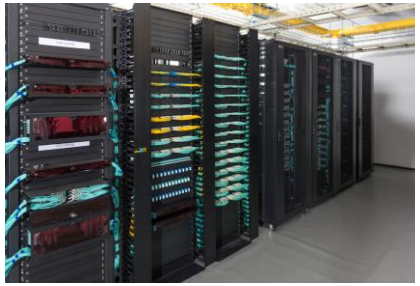
Index

- Introduction
- Existing Challenges
- Related Works
- Proposed Solution
- Scope of the Work
- Methodology
- Implementation
- Experiments and Results
- Discussion
- Conclusion
- Future Works

Introduction

- Recent trends in edge computing

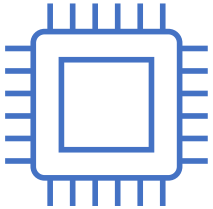
- Robustness of cloud computing



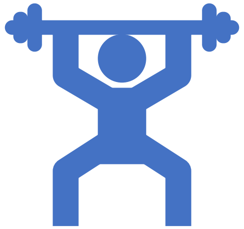
- Leveraging edge and cloud for machine learning

Existing Challenges

Viewpoint problem with cloud training, edge inference



Resource constraints of edge devices for on-device training



How to leverage the strengths of both types of devices???

Related Works

Edge-Cloud Collaboration

- Big data driven edge-cloud collaboration architecture (Yang 2020)
- Co-edge (Hu et al. 2020)

Federated learning

- Federated learning in resource constrained edge devices (Wang et al. 2019)
- Federated learning for edge networks (Khan et al. 2020)
- Secure and efficient federated learning for smart grid with edge-cloud collaboration (Su et al. 2021)

Our Proposed Solution

Edge and Cloud can 'collaborate' with each other

Proximity of edge to data source

Hardware capabilities of cloud computing

Pre-training on edge with local data

Fine tuning on server with global data

Scope of the Work

We propose ECAvg as a collaborative edge-cloud approach

Edge devices for pre-training

Cloud for model aggregation and fine-tuning

3 experiments, 3 different architectures:

CIFAR10

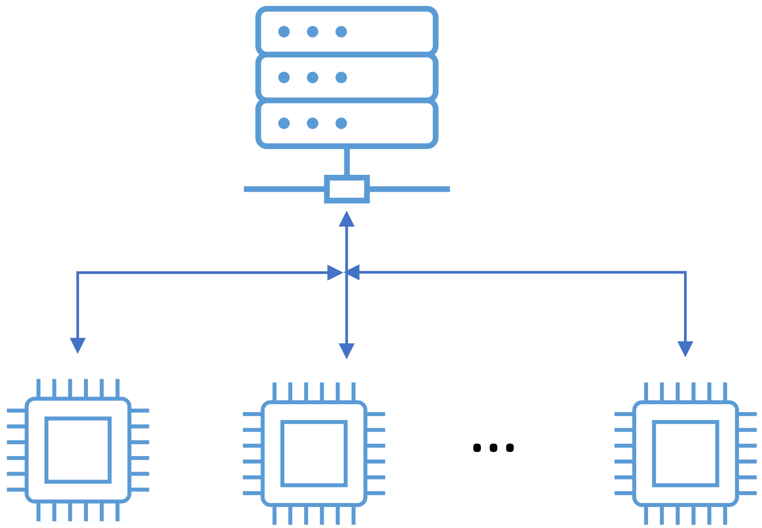
CIFAR100

MNIST

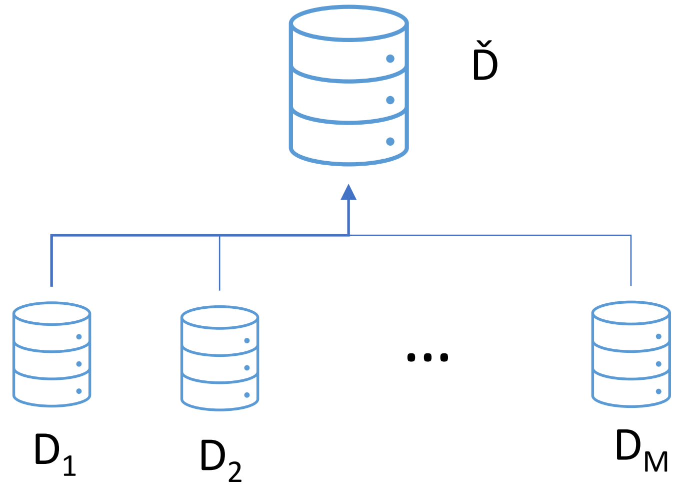
Explore the role of transfer learning in our approach

Methodology

- Environments:
  - Central server
  - and M client edge devices



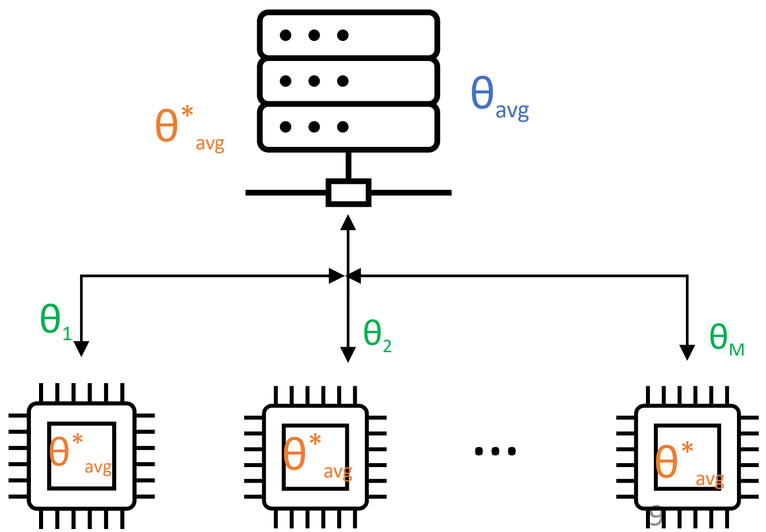
- Data:
  - D_i for each i client
  - Ď aggregated on server

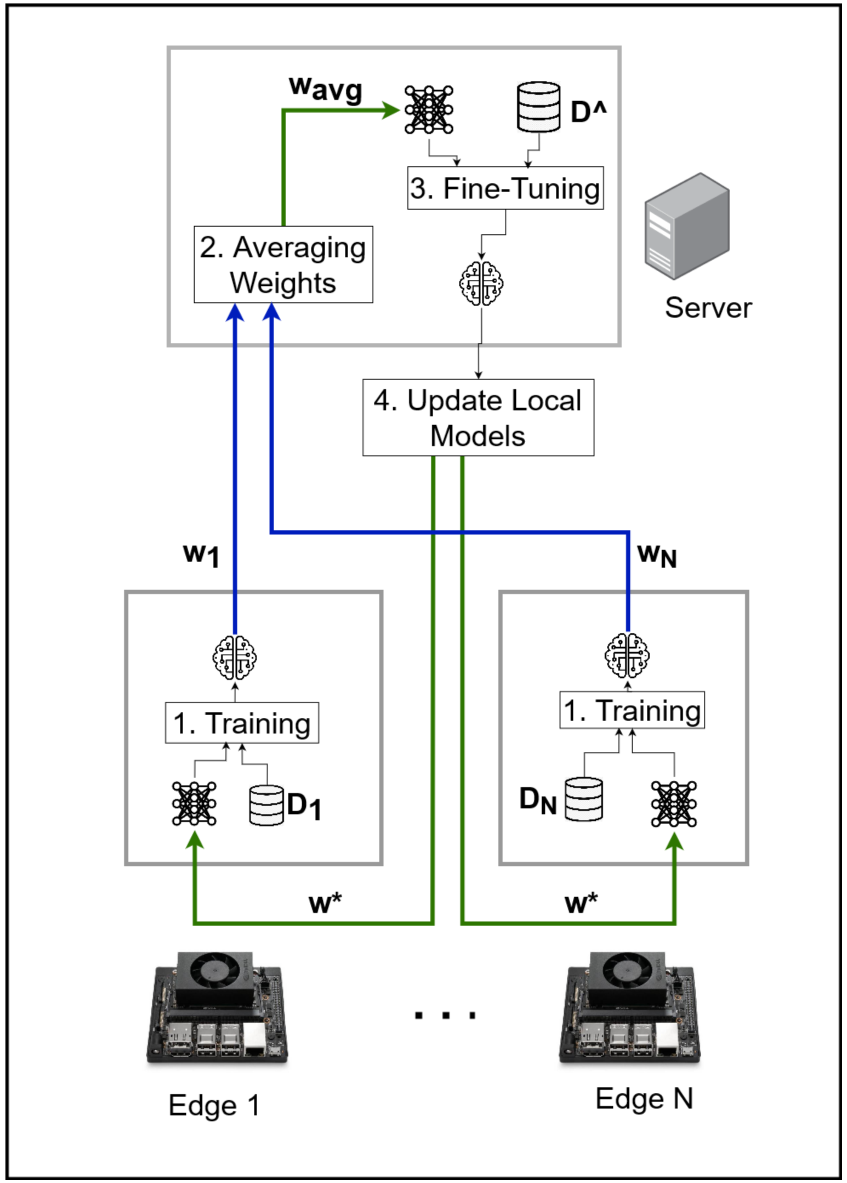


Methodology

Training:

- Each client computes the parameters θ_i on its local dataset D_i and sends to the server
- Server averages the parameters into a global model h_{avg} with parameters θ_{avg}
- θ_{avg} are fine-tuned on the global dataset into θ^*_{avg} and updated to the client models
- Client devices retrain on data




wavg
D^
3. Fine-Tuning
2. Averaging Weights
Server
4. Update Local Models
w1
wN
1. Training
D1
1. Training
DN
w*
w*
. . .
Edge 1
Edge N

Implementation

Setup:

Desktop as server device

Two A203 Mini PC (developed upon Jetson Xavier NX) client devices

Datasets:

D_1 on $Edge_1$ and D_2 on $Edge_2$

Models:

Two identical client models, M_1 and M_2

Server model M built from averaged M_1 and M_2 weights

Implementation

Training

M_1 and M_2 trained on D_1 and D_2 respectively on the edge

Weights of M_1 and M_2 averaged into a server model M

M finetuned on aggregate data Ď

Fine-tuned weights of M updated on edge models M_1 and M_2

M_1 and M_2 retrained on respective data

3 separate experiments

Experiment 1: MobileNetV2 on CIFAR10

Experiment 2: ResNet50 on CIFAR100

Experiment 3: Simple neural net with one hidden layer on MNIST

Experiment 1

Datasets:

CIFAR10 split into two non-overlapping sets, each with 5 classes

One set per client device

Complete CIFAR10 dataset on server

Models:

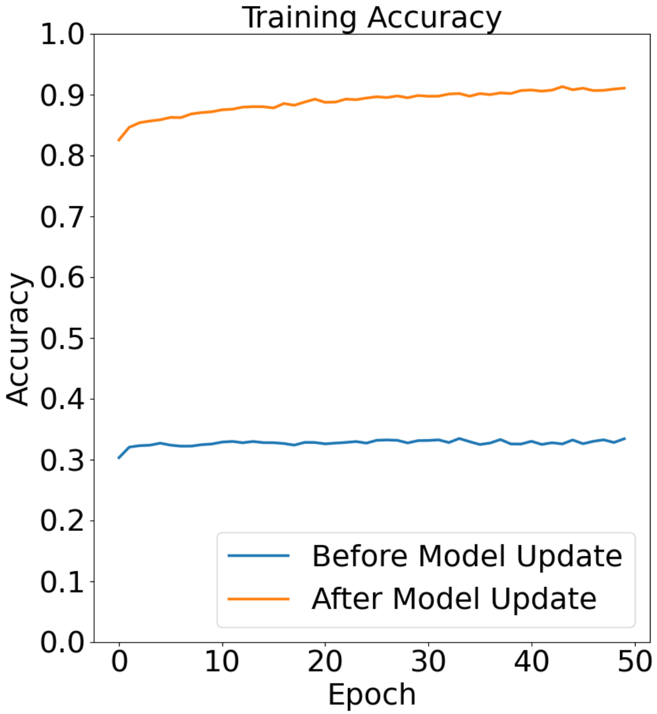
Two identical **MobileNetV2** classifiers pretrained on ImageNet

Training

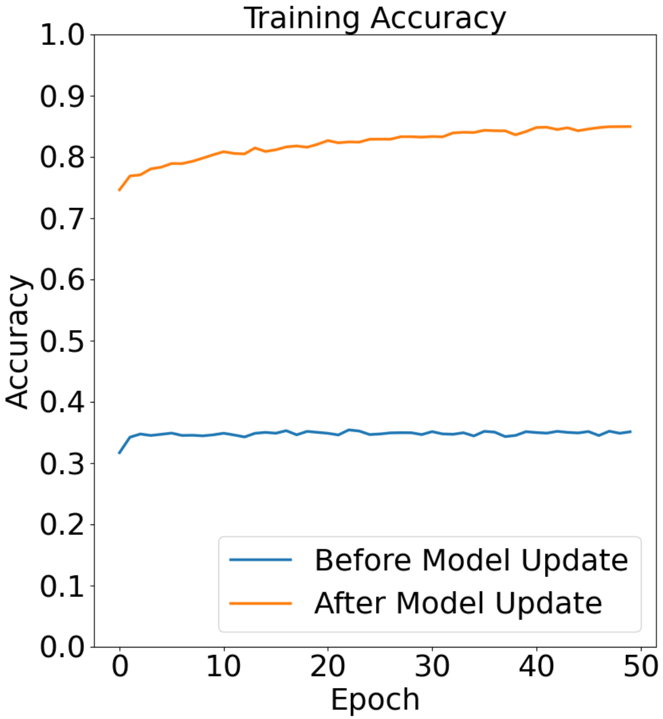
As described in Implementation

Experiment 1

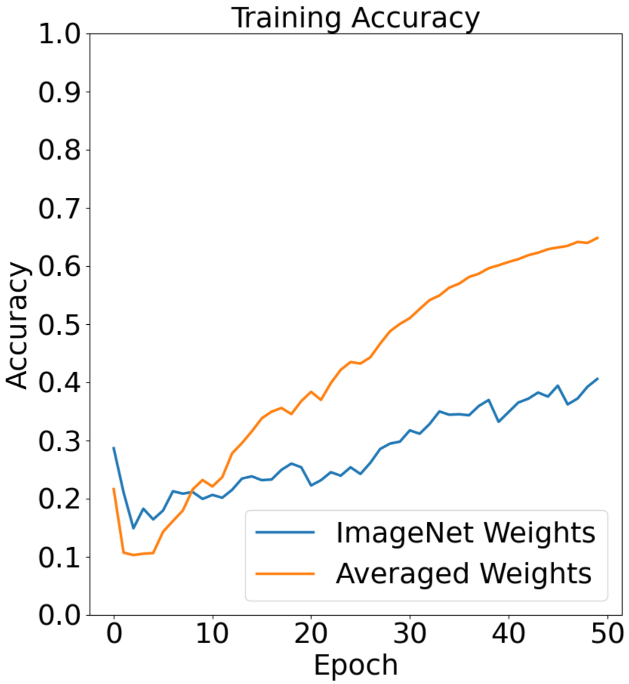
- Results



Edge 1



Edge 2



Server

Experiment 1

- Testing Results

Device	Setup	Acc	Precision	Recall	F1 Score
Edge 1	Before Update	0.3886	0.4096	0.3886	0.3597
Edge 1	After Update	**0.8116**	**0.8134**	**0.8116**	**0.8090**
Edge 2	Before Update	0.3688	0.3809	0.3688	0.3641
Edge 2	After Update	**0.8740**	**0.8862**	**0.8740**	**0.8711**
Server	ImageNet weights	0.3660	0.4096	0.3660	0.3203
Server	Averaged weights	**0.6696**	**0.6946**	**0.6696**	**0.6641**

Experiment 2

Datasets:

CIFAR100 split into two non-overlapping sets, each with 50 classes

One set per client device

Complete CIFAR100 dataset on server

Models:

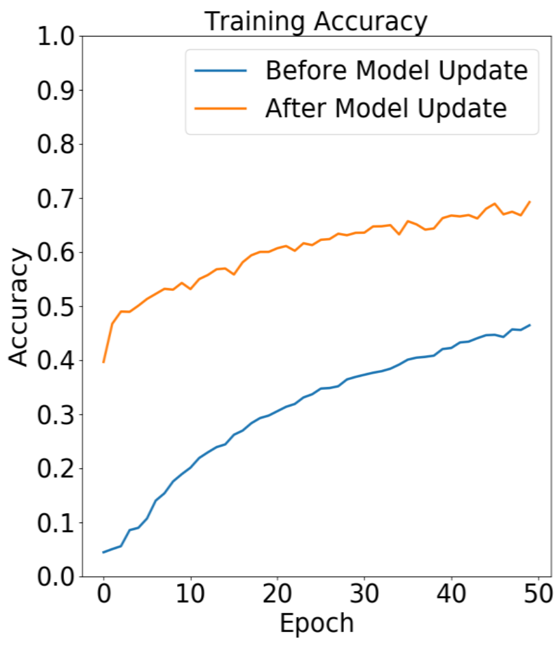
Two identical ResNet50 classifiers pretrained on ImageNet

Training

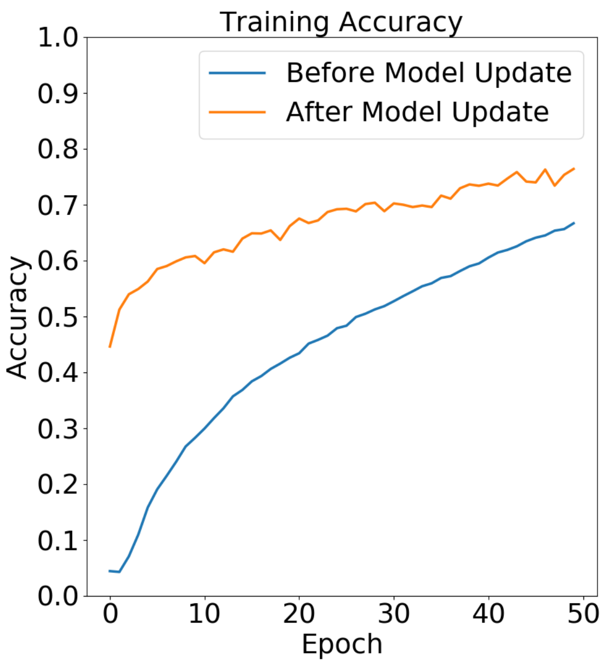
As described in Implementation

Experiment 2

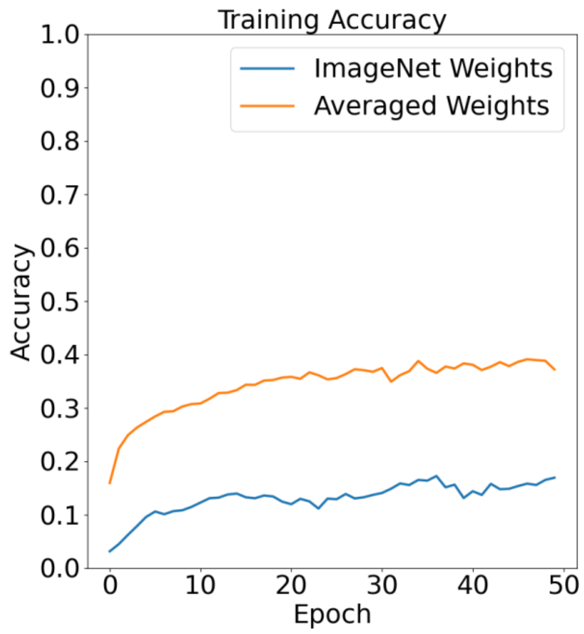
- Results



Edge 1



Edge 2



Server

Experiment 2

- Testing Results

Device	Setup	Acc	Precision	Recall	F1 Score
Edge 1	Before Update	0.3964	0.4441	0.3964	0.3927
Edge 1	After Update	**0.5156**	**0.5355**	**0.5156**	**0.5127**
Edge 2	Before Update	0.4878	0.4955	0.4878	0.4809
Edge 2	After Update	**0.5180**	**0.5308**	**0.5180**	**0.5140**
Server	ImageNet weights	0.2100	0.2076	0.2100	0.1910
Server	Averaged weights	**0.3745**	**0.3938**	**0.3745**	**0.3639**

Experiment 3

Datasets:

MNIST split into two non-overlapping sets, each with 5 classes

One set per client device

Complete MNIST dataset on server

Models:

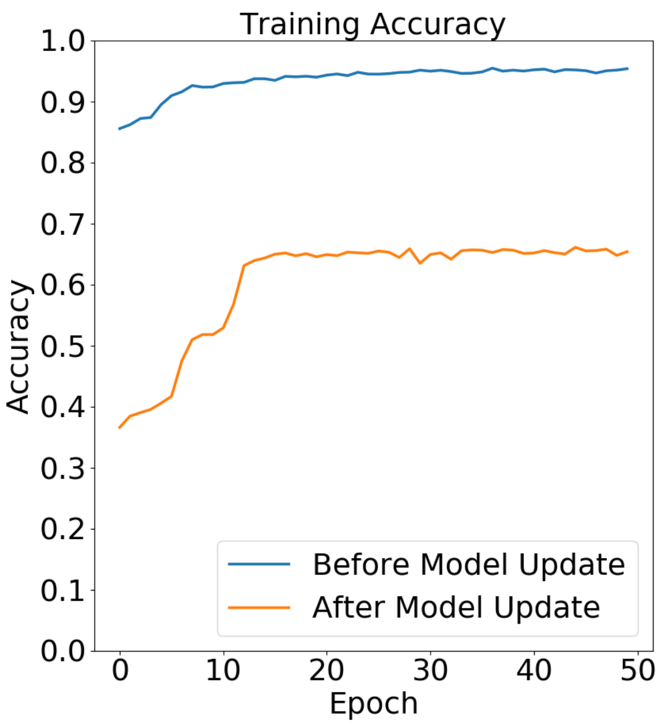
Two identical simple neural networks classifiers

Training

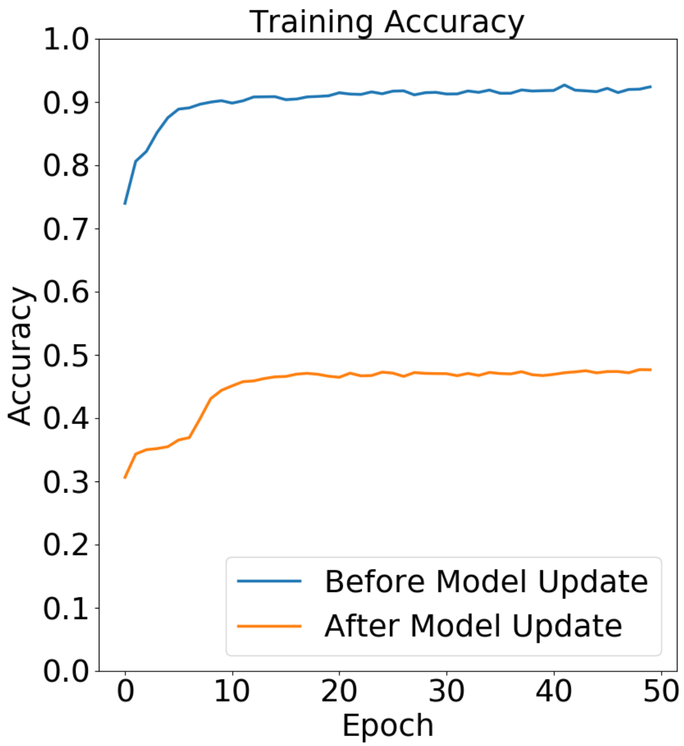
As described in Implementation

Experiment 3

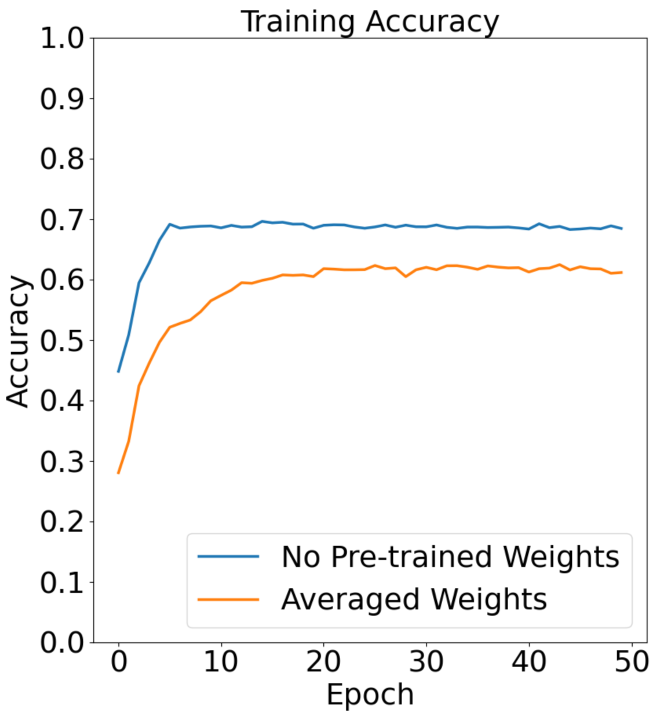
- Results



Edge 1



Edge 2



Server

Experiment 3

- Testing Results

Device	Setup	Acc	Precision	Recall	F1 Score
Edge 1	Before Update	**0.9722**	**0.9737**	**0.9722**	**0.9725**
Edge 1	After Update	0.7544	0.6784	0.7544	0.6980
Edge 2	Before Update	**0.9568**	**0.9594**	**0.9568**	**0.9571**
Edge 2	After Update	0.5466	0.4836	0.5466	0.4617
Server	No pre-training	**0.7938**	**0.8128**	**0.7938**	**0.7854**
Server	Averaged weights	0.6417	0.6854	0.6417	0.6285

Discussion

Performance improvements

Improved generalisability by fine-tuning

Positive transfer of knowledge between models

Task similarity between local and global datasets

Negative transfer learning with simple networks

Performance loss due to negative transfer learning

Avoided in Exp 1 and Exp 2 due to complex model architectures and regularization

Neural network in Exp 3 lacked these benefits and thus negative transfer learning occurred

Conclusion

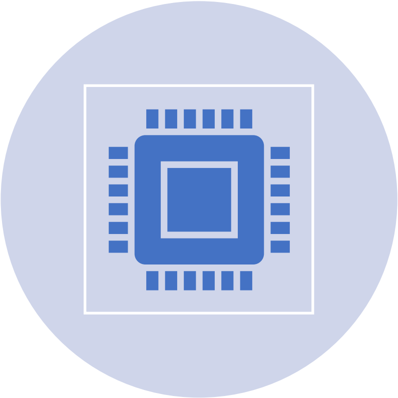
Edge and cloud can complement each other in through a collaborative paradigm

ECAvg as a collaborative learning approach

Performance improvements observed when using deep neural network architectures

Decrease in performance for simple architectures

Future Works



INCLUDING MORE CLIENT EDGE DEVICES

VARYING THE DATASET SPLIT RATIO

FUTURE APPLICATIONS

Thank you

ANALYTICS EVERYWHERE LAB

Dr Hung Cao
Assistant Professor
Lab Director
Analytics Everywhere Lab
University Of New
Brunswick, Canada
hcao3@unb.ca

Dr Francis Palma
Assistant Professor
Faculty of Computer Science
University Of New
Brunswick, Canada

Dr Monica Wachowicz
Adjunct Professor
Associate Dean
Geospatial Science
RMIT University, Australia

Dr Trevor Hanson
Professor
Faculty of Civil Engineering
University Of New
Brunswick, Canada

Asfia Kawnine
MSc Student

Atah Nuh Mih
MSc Student

Alireza Rahimi
MSc Student

Simran Dadhich
MSc Student

Arman Nik Khah
MSc Student

Connor Kenneth
McLenaghan
Undergrad Student

…