

Efficient-Edge Computing Federated Learning for Industrial Internet-of-Things

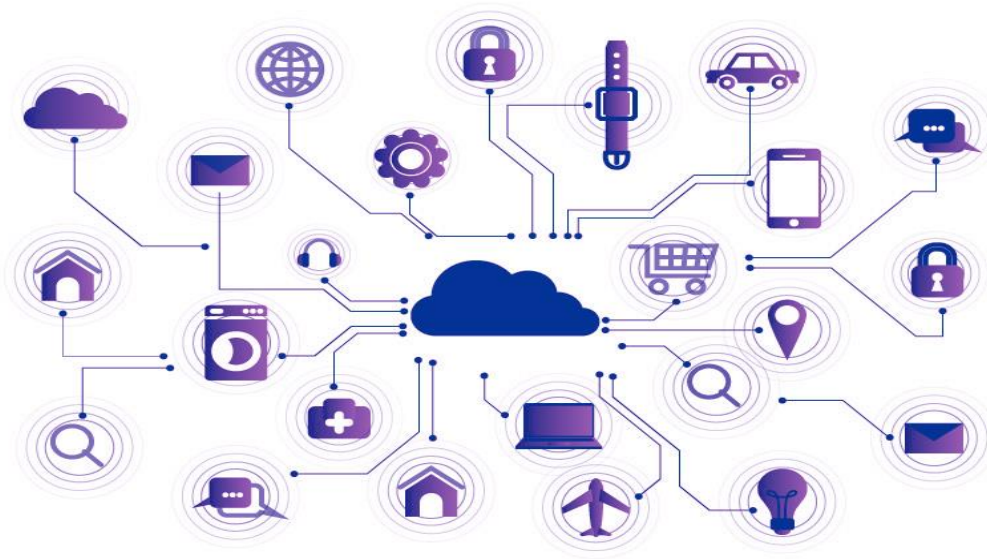
Vidya Valsalan ,M.Tech CSE-IS

Admission Number :19MT0434

Department Of Computer Science & Engineering

Introduction

- ▶ Internet Of Things is a platform where physical object “things” are connected to the internet ,So they can interact ,collabarate and exchange data with each other.
- ▶ From any time,any place connectivity to anyone,we will now have connectivity for anything.



Introduction

- ▶ Huge Amount of smart devices connected together in IoT, we are able to get access to massive user data.
- ▶ Cloud Computing is a model for enabling convenient ,on-demand network access to a shared pool of configurable computing resources.
- ▶ Ubiquitous ,dynamic and on-demand access.
- ▶ More efficient resource utilization,usage based costing.

Intoduction

- ▶ Billions of phones & IoT devices constantly generating data. Data enables better products and smarter model.
- ▶ Train task-specified machine learning models and utimately provide high-quality smart services and products.
- ▶ Collect scattered user data to a central cloud for modeling and then transfer the trained model to user devices for task inferences is not an efficient way.
- ▶ User-sensitive data are required to upload to the remote cloud, it may impose great privacy leakage risk

Motivation

- ▶ Much of the data is born decentralized. Data may be too sensitive or large in size so it is preferable not to log it to the data center purely for the purpose of model training.
- ▶ Data transmission and model transfer will result high communication cost.
- ▶ Machine Learning on decentralized data. Enable edge devices to do state of the art machine learning without centralizing data and with privacy by default.
- ▶ Train Model from device directly, no need to send privacy information to server.

Existing Approach

- ▶ Approach 1: Performing Federated Learning for all clients with all n data samples (Gradient Descent).
- ▶ More computational power and need to transfer more parameter it leads to more communication cost.
- ▶ Approach 2: Performing Federated Learning for a randomly selected client that has n_k training data samples.
- ▶ A single step of gradient decent is done per round
- ▶ More communication round it leads more communication cost.

Proposed Approach

- ▶ Performing Federated Learning for fraction of clients with batch B data samples(Mini Batch Gradient Descent).
- ▶ Learning rate: η , total number of samples: n , total number of clients: k , Number of samples on a client k : n_k .
- ▶ In a round t :
 - ▶ The central server broadcasts current model w_t to each client; each client k computes gradient: g_k ,on its local data.
 - ▶ Approach: Each client k computes it's weight
 - ▶ $w_{t+1}^k \leftarrow w_t - \eta g_k$
 - ▶ Central Server Perform the aggregation :
 - ▶ $w_{t+1} \leftarrow \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k$

Proposed Approach

Algorithm Server Execute:

Initialize the global model w_0 by averaging each local

for each round $t=1,2,\dots$ do

$m \leftarrow \max(C.K, 1)$

$S_t \leftarrow$ set of clients that send the model

for each client $k \in S_t$ in parallel do

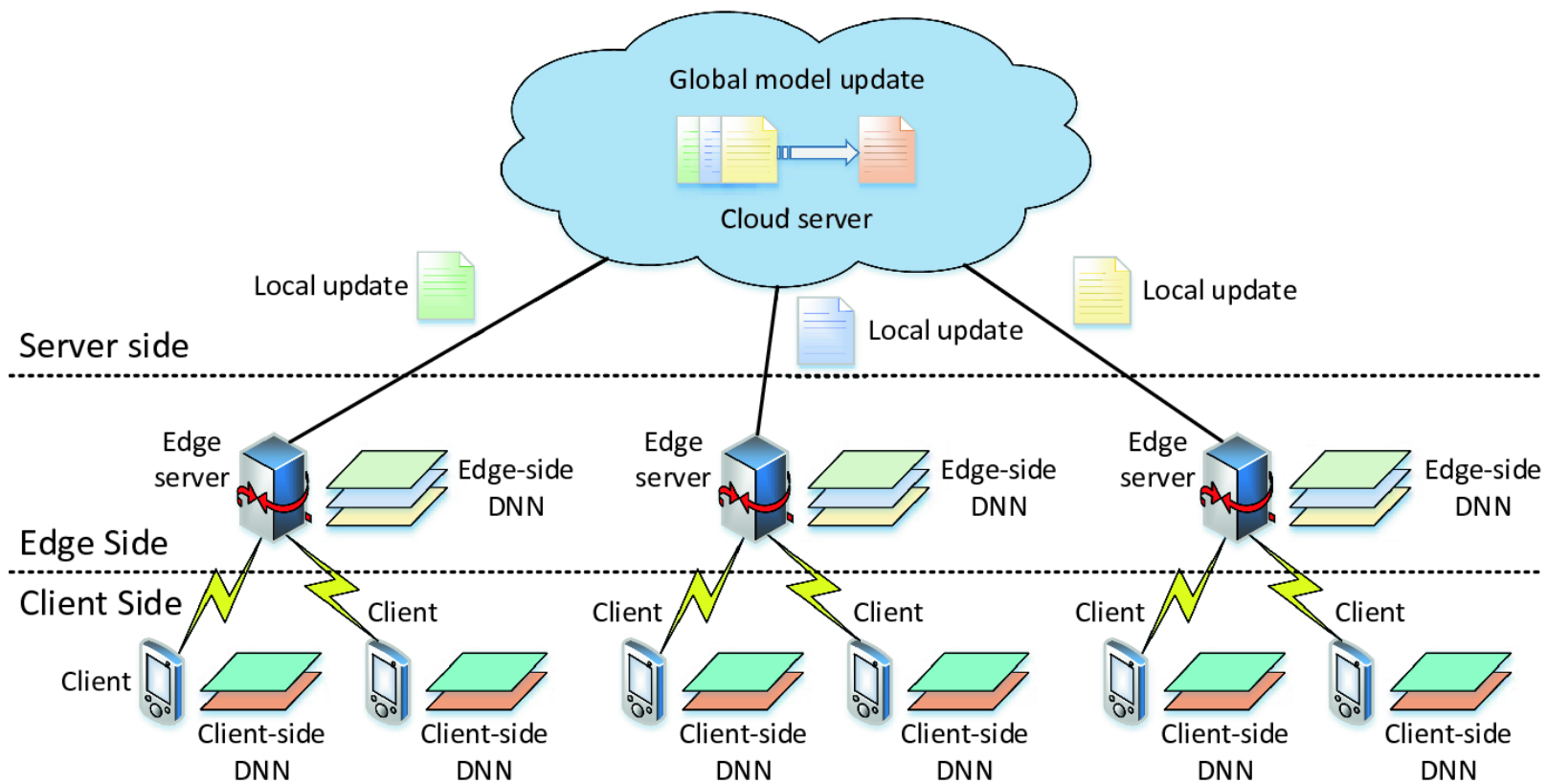
$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$

$w_{t+1} \leftarrow \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k$

Proposed Approach

- ▶ Algorithm: Client Execute
- ▶ Input: Global update of last round T^t .
- ▶ 1. Initialize the global model M_0 by averaging each local model.
- ▶ 2. for each iteration do
- ▶ 3. for each user $u_i \in U$ do
- ▶ 4. u_i train its local model on its data D_i
- ▶ 5. u_i calculates its updated weight W_i .
- ▶ 6. If current updated weight varies more than the threshold value. update its local model for global aggregation. Otherwise u_i continues its local training
- ▶ 7. end for
- ▶ 8. end for

System Model



Implementation and Results

- ▶ Used MNIST digit recognition dataset contains 70000 handwritten grayscale digits images ranging from 0 to 9.
- ▶ Mainly 4 components:
 - ▶ 1) A server-to-client broadcast step (*broadcast_fn()*)
 - ▶ 2) A local client update step (*client_update()*)
 - ▶ 3) A client-to-server upload step
 - ▶ 4) A server update step. (*server_update()*)

Implementation and Results

Number Of Communication Round	FedSGD	FedAvg with Batch Size=10	FedAvg with Batch Size=10
50	0.754	0.803	0.786
100	0.762	0.836	0.810
200	0.798	0.869	0.838

Literature Survey

Title	Cloud Machine Learning Model Selection for IIoT and Smart City services.
Author	Basheer Qolomany, Ihab Mohammed
Problem Definition	cloud service providers collect big data from resource-constrained devices for building Trust based Cloud ML-based prediction models that are then sent back to be run locally on the intermittently-connected resource-constrained devices
Methodology	ML models from which a subset needs to be selected and deployed on IoT devices for T time slots.
Limitations	Centralization of data

Literature Survey

Title	Personalized Federated Learning for Intelligent IoT Applications
Author	Qiong Wu,Kaiwen He,Xu Chen
Problem Definition	personalized federated learning framework in a cloud-edge architecture for intelligent IoT applications. To cope with the heterogeneity issues in IoT environments.
Methodology	Federated Stochastic gradient descent(SGD) send the gradient to server.
Limitations	More communication and more number of rounds.

Conclusion

- ▶ Federated Learning can be a promising solution that enables on-device machine learning without the need to migrate the private end-user data to a central cloud.
- ▶ Fed Avg Model trains high-quality model using relative few rounds of communication.
- ▶ It provide cloud edge architecture for intelligent IoT application with more data privacy.

THANK YOU