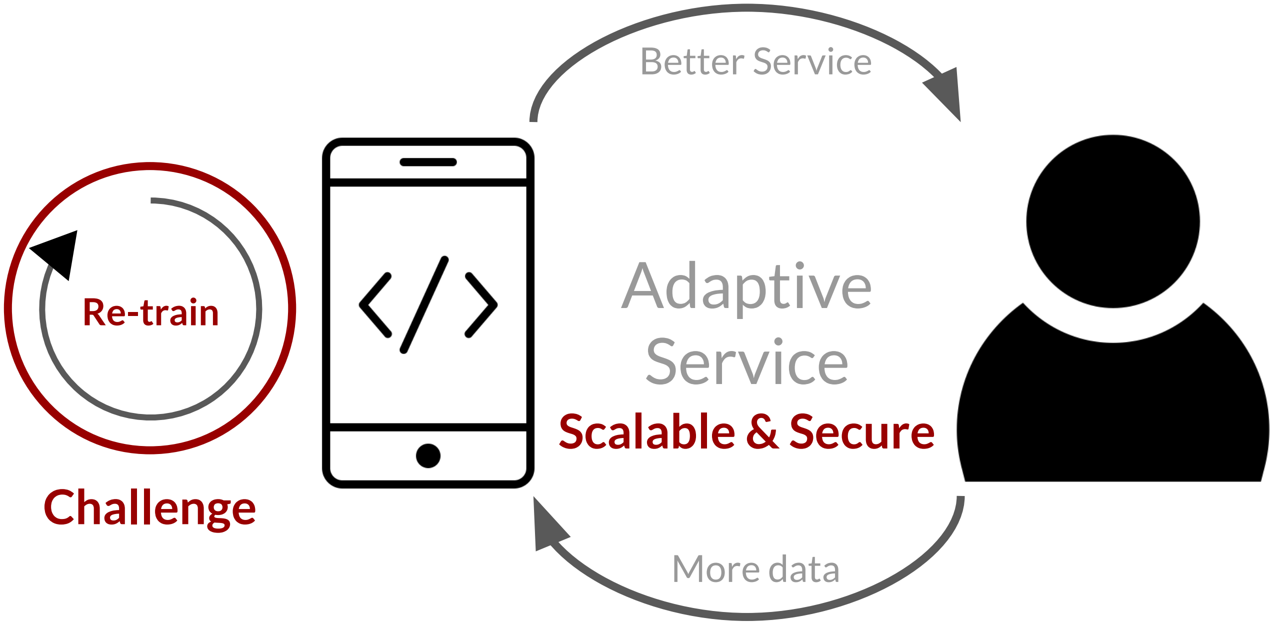
Mobile & Ubiquitous Computing, 2020 Spring

FedEx: Scalable Framework for Mobile User-Adaptive Deep Learning Training

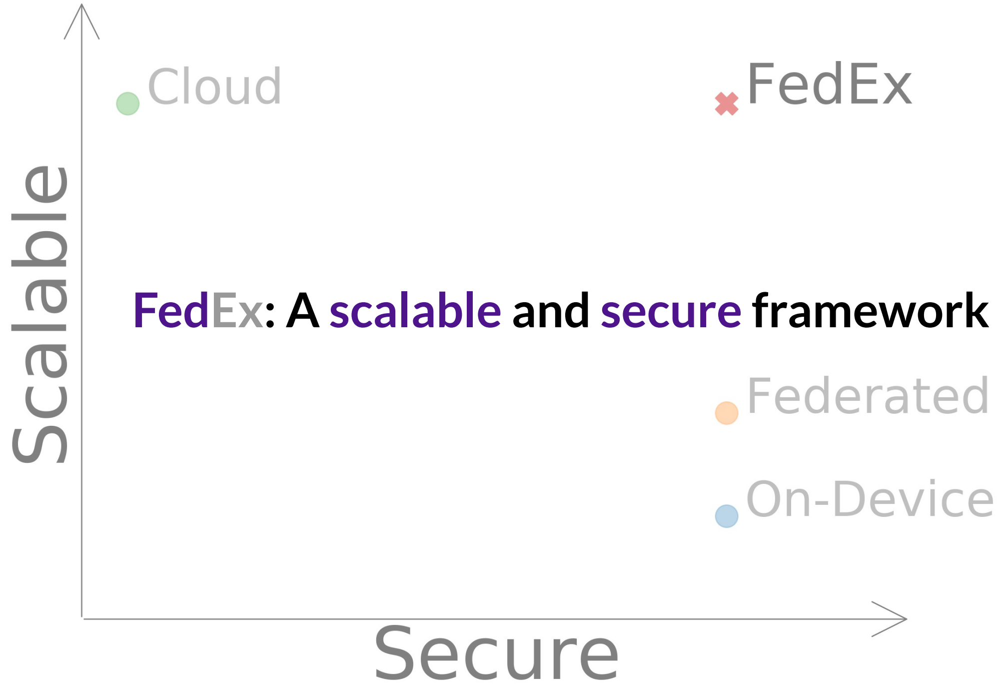
{Ahnjae Shin, Kyunggeun Lee, Heeseung Yun, Taebum Kim} @ Team 1

Recap


Better Service
Re-train
Challenge
Adaptive
Service
Scalable & Secure
More data

Scalable Able to scale to large models

Secure Able to train without raw data

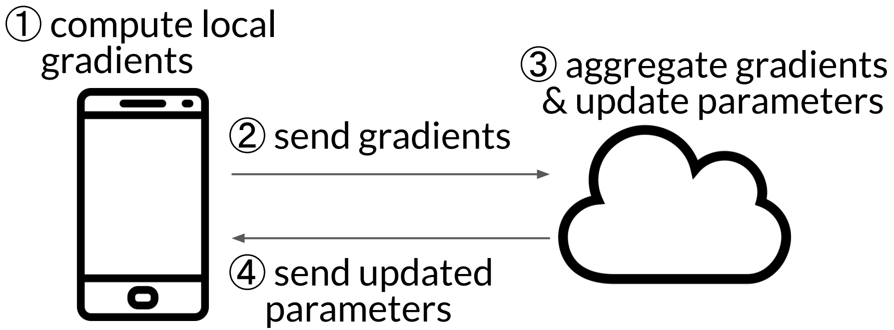

Cloud
FedEx
FedEx: A scalable and secure framework
Federated
On-Device
Scalable
Secure

We ship fresh models daily

Federated Learning

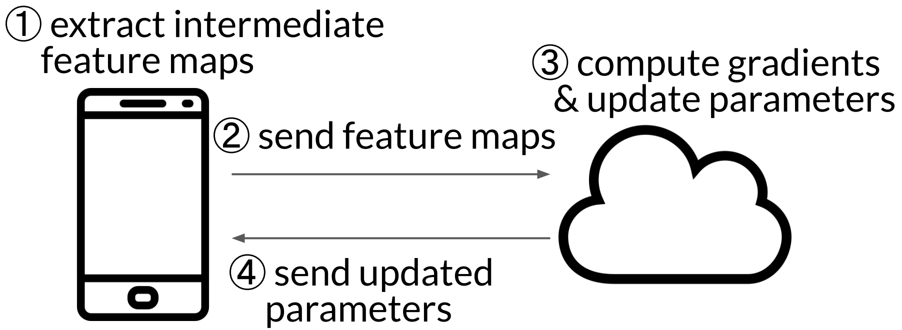
FedAvg

- Compute in mobile devices, update aggregated gradients in clouds
- Computation: mobile device
- Communication rounds: O(NP)



FedEx

- Extract intermediate feature maps in mobile devices, train in clouds
- Computation: cloud
- Communication rounds: constant



McMahan et al. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In AISTATS

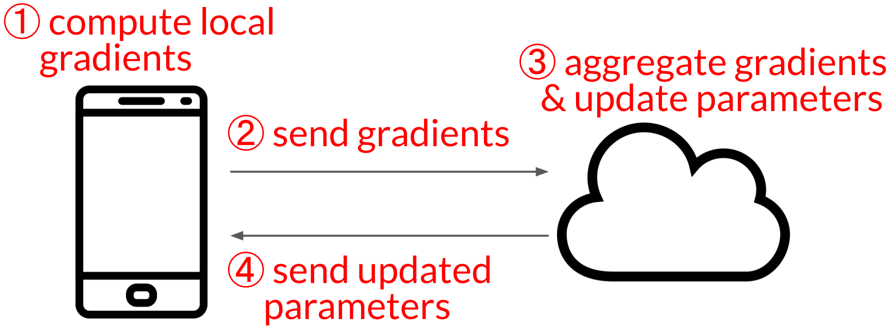
Federated Learning

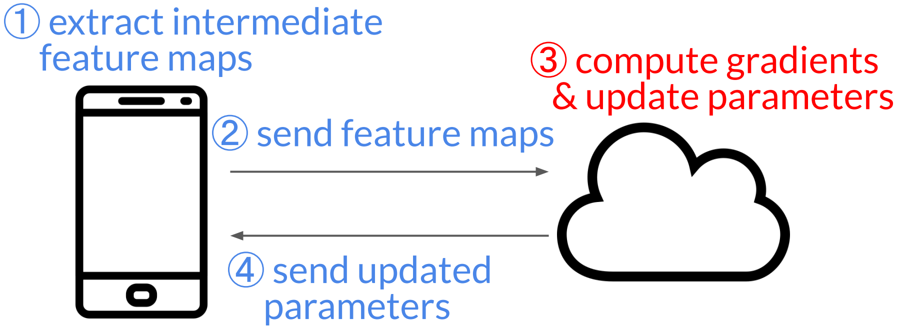
FedAvg

- Compute in mobile devices, update aggregated gradients in clouds
- Computation: mobile device
- Communication rounds: O(NP)

FedEx

- Extract intermediate feature maps in mobile devices, train in clouds
- Computation: cloud
- Communication rounds: constant





: occurs every step

: occurs only once

McMahan et al. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In AISTATS

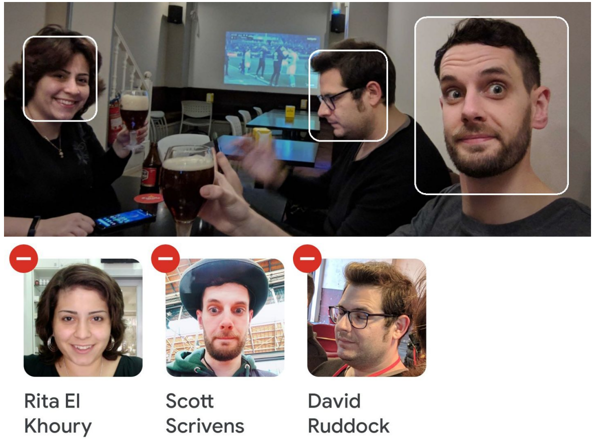
Refinement

Refinement

- Problem statement
  - Existing federated learning (ex. FedAvg) is slow
    - repeated computation in mobile devices
    - heavy communication between mobile devices and clouds.
- Project main idea
  - Extract intermediate feature maps in mobile devices, and train in cloud servers.
    - Only constant computation in mobile devices.
    - No raw data is sent to the server.
- Project scope
  - FedEx outperforms FedAvg for a federated transfer learning task w.r.t. training throughput.
- **Target application (changed)**

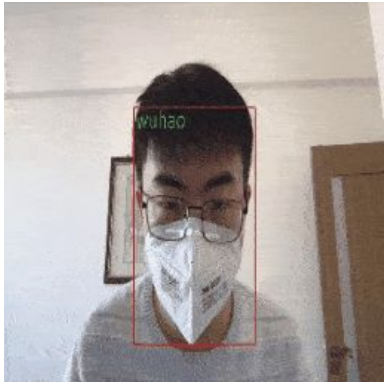
Target application

- Handwriting → **Face recognition**
  - Motivation
    - Necessity of additional training
    - No sequential modeling, moderate complexity
    - More diverse distribution across a variety of users



Target application

- Handwriting → **Face recognition**
  - After literature review, we decided to focus on a more intriguing problem
  - Some open questions in face recognition domain
    - Technology of the moment: [masked face recognition]
    - Unsupervised face clustering
      (ex. How many characters are there in the smartphone gallery?)



Wang et al. (2020). Masked Face Recognition Dataset and Application. *arXiv preprint arXiv:2003.09093*.
Tapaswi et al. (2019). Video Face Clustering with Unknown Number of Clusters. In ICCV.

Target application

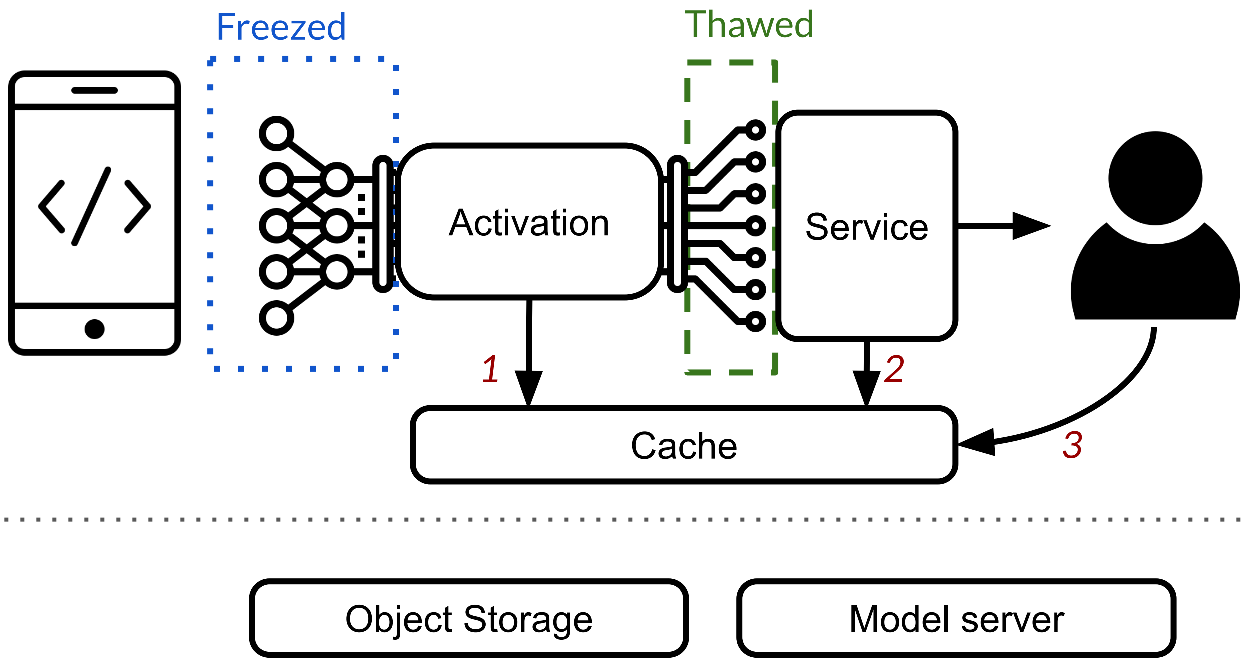
- Handwriting → **Face recognition**
  - After literature review, we decided to focus on a more intriguing problem
  - Some open questions in face recognition domain
    - Technology of the moment: masked face recognition
    - Unsupervised face clustering
      (ex. How many characters are there in the smartphone gallery?)
  - Objective
    - For given set of facial images from a few characters, train a model that can discern "Who is who" by leveraging user-adaptive training framework of FedEx

Wang et al. (2020). Masked Face Recognition Dataset and Application. *arXiv preprint arXiv:2003.09093*.
Tapaswi et al. (2019). Video Face Clustering with Unknown Number of Clusters. In ICCV.

Architecture

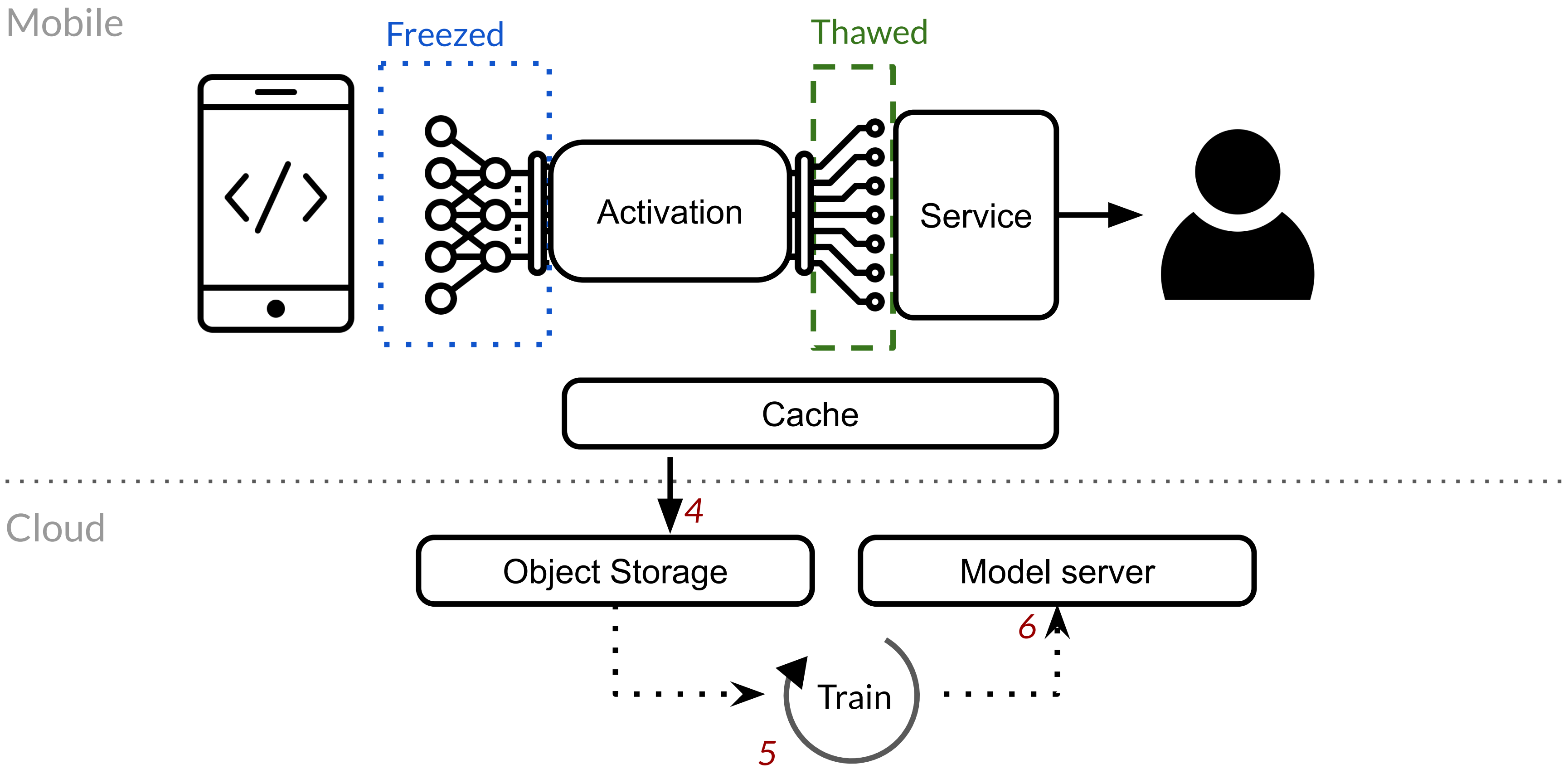
System Architecture

Mobile

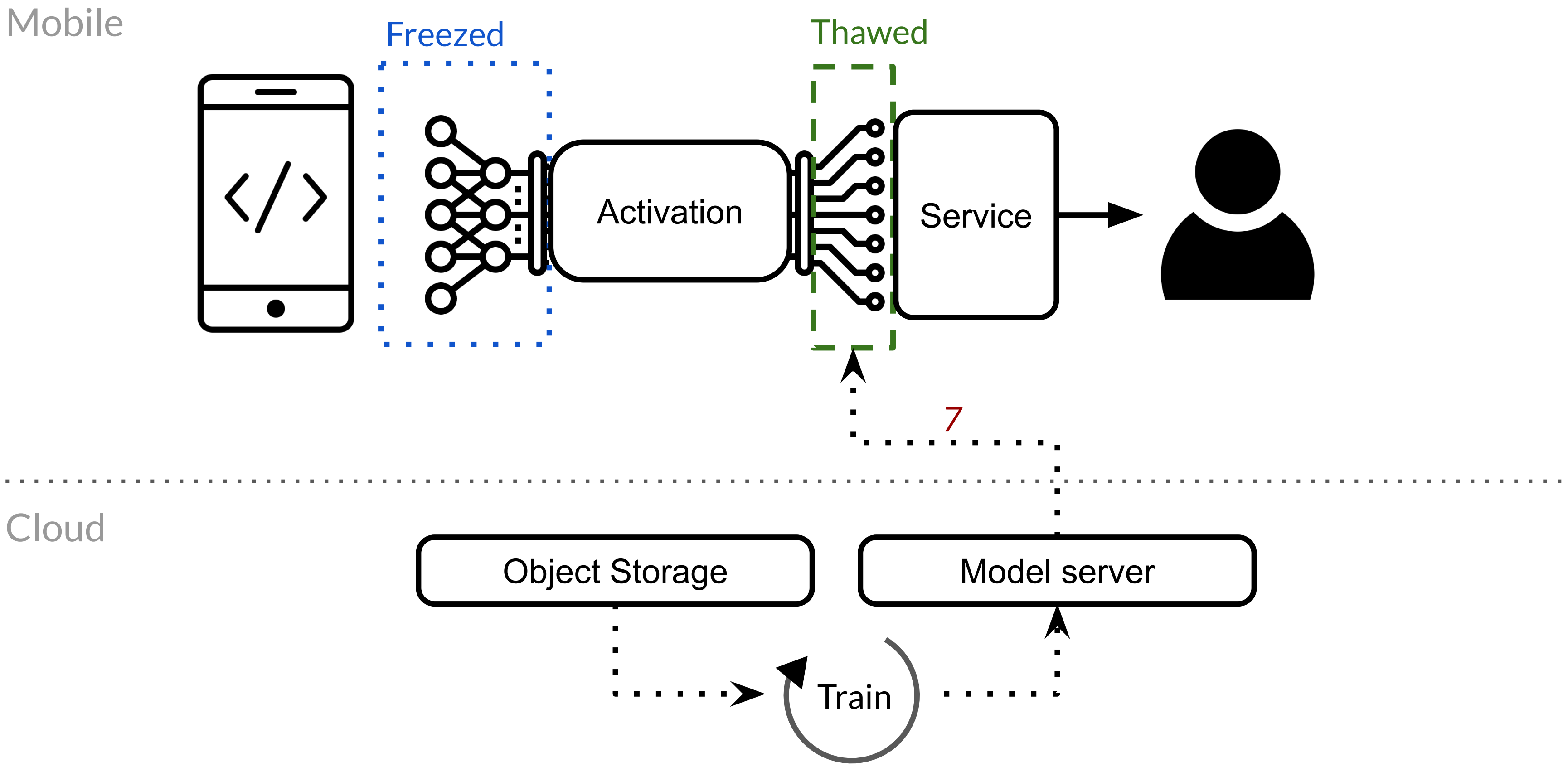


Cloud

System Architecture



System Architecture



Project Status

- Features
    - Basic application with TFLite models
    - Deploy TFLite models that output intermediate feature maps
    - Partial training using intermediate feature maps
    - Client-server communication (feature maps & parameters)
    - Face recognition application
    - Personalized data collection in app
- Evaluation
    - Deploy light model
    - Deploy heavier models
    - Implement FedAvg
    - Measure training throughput
    - Measure device energy consumption

Demo

Demo

1. Run app that inference DL models
2. The app will save **intermediate outputs as file**
3. Use adb (Android debug bridge) to copy files to cloud
4. Use intermediate outputs to **train model**

Remaining Features

1. Run an app with DL inference models
2. The app will ~~save **intermediate outputs as file**~~ send files to cloud
3. ~~Use adb (Android debug bridge) to copy files to cloud~~
4. Use intermediate outputs to **train model**
5. **Pull new model from cloud**

Challenges & Solutions

Remaining Challenges

- Personalized data collection
  - Should prove that additional training improves the accuracy.
  - However, a significant amount of labeled data is required.
- Implementation of baseline system (FedAvg)
  - The current DL frameworks (TensorFlow, PyTorch, etc.) do not support Python APIs for federated learning.
- Large scale experiment
  - As the number of involved mobile devices grows, FedEx is expected to outperform FedAvg by a larger gap.
  - In reality, however, it is difficult to conduct a very large-scale experiment with a great number of mobile devices.

Possible Solutions

- Personalized data collection
  - Tons of massive annotated datasets available: CelebA, VGGFace2, TrillionPairs, etc.
  - Split the existing dataset for:
    - Primary training
    - Secondary training (sets of multiple identities)
  - It is also possible to leverage YouTube videos by applying face tracking
- Implementation of baseline system (FedAvg)
  - On-device training using Tensorflow Java APIs
  - Simulate mobile devices on desktops (TensorFlow Federated)
- Large scale experiment
  - Simulate mobile devices on desktops (mocking behavior)

Liu et al. (2015). Deep Learning Face Attributes in the Wild. In ICCV.
Cao et al. (2018). VGGFace2: A dataset for recognising face across pose and age. In AFGR.

Schedule

Overall Plan

#Iter.	Objective	Duration	Misc.
✓	**Ideation & Design** Brainstorming, Literature Review, Proposal Feedback	4/1 - 4/14	4/6: Proposal
✓	**System & Architecture** Cloud Setup, System Design, Client Mockup	4/15 - 4/28	
✓	**Model & Algorithm** Data Preparation, Model Implementation, Integration	4/29 - 5/12	5/11: Project Demo
4	**Optimization & Experiment** Ablation Studies, Alpha Deployment, Load Testing	5/13 - 5/26	
5	**Deployment & Deliverable**	5/27 - 6/8	6/8: Final Presentation

Task Assignment

Task	Ahnjae Shin	Kyunggeun Lee	Heeseung Yun	Taebum Kim
Simple client app implementation	O		O	
Server basic implementation		O		O
Light-weight TFLite model implementation & deployment		O	O	O
Face recognition model development	O		O	O
Face recognition app implementation		O	O	O
Baseline solution (FedAvg) implementation & experiment	O	O		
Integrate server and client	O	O	O	O

Deliverable

Deliverable

- Final Deliverable
  - Technical report regarding evaluations
  - Demo application
- Success Criteria
  - Our method succeeds to **train larger model faster** than baseline method
  - Our method succeeds to **use less energy** than baseline method
- Potential Threats
  - Absence/lack of TFLite Android API
  - *Procrastination*

Lesson Learned

- Android application development
  - Gradle build configuration is hard — Android Studio + Gradle seems to have an OS-dependent issue (in Ubuntu).
  - Android application programming/debugging process
- TFLite deployment
  - TFLite converter has become much more powerful and convenient in TF2.x.
  - Yet, subtle issues seem to be still present in TFLite converter (accuracy problem).
- On-device training
  - The field of on-device (or federated) training is still very unexplored.
  - Difficulties in finding reference materials
  - Our project is expected to run into many unexpected issues implementing on-device training.

We ship fresh models daily

Thank you

Backup Slides

Remaining Todos

1. Features
   a. Device-cloud communication: save/load models and features automatically
   b. Label collection: collect personalized labels in our app
2. Evaluation
   a. Implement & evaluate on relatively heavier models
      (currently only implemented a light-weight sample model (MobileNet) & task)
   b. Implement or simulate baseline solution (FedAvg)