Meta-HAR: Federated Representation Learning for Human Activity Recognition

Xiaobing Chen
August 14, 2021

Outlines

1. Introduction

- Human Activity Recognition (HAR)
- Limitations of Federated learning (FL) for HAR

2. Meta-HAR Method

- Embedding Network
- Personalized Classification

3. Experiments

- Dataset Collection and Processing
- Implements
- Results

4. Conclusions and Discussion

1 Introduction

Introduction

- **Human Activity Recognition (HAR)**: a task of recognizing human activity types based on mobile sensor data, such as accelerometer and gyroscope.
- **Neural Network on HAR**: replace manual feature engineering; great acc improvement.
- **Concerns of Centralized Learning**: privacy restrictions on users' data collection.
- **Federated Learning on HAR**: utilized users' sources; protect privacy.
- **Challenges for applying Federated Learning (FL) to HAR [1]**:
  1. heterogeneity in label distribution across users
  2. heterogeneity in the signal distributions of the same activity across users
  3. The generalization limitation of FL to new user

[1]. Li, Chenglin, et al. "Meta-HAR: Federated Representation Learning for Human Activity Recognition." Proceedings of the Web Conference 2021. 2021.

Introduction-heterogeneity in the signal distributions

- **Heterogeneity in Human Activity Recognition (HAR)**: HHAR dataset collects accelerometer and gyroscope data from 9 users with different smart devices to predict 6 activities.

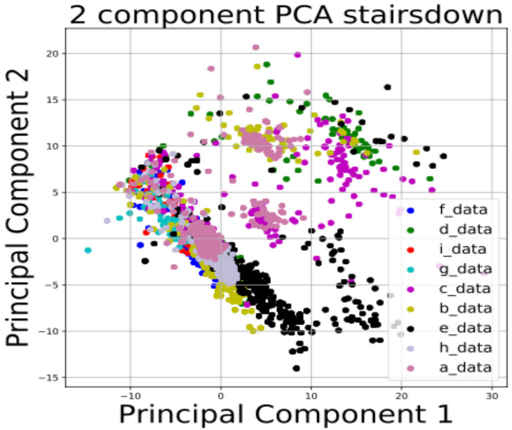

Figure 1: The data of "stairsdown" from the HHAR dataset after reducing dimension to 2D using PCA. There exists a clear cluster relationship among different subjects' data and also heterogeneity in the signal distributions.

Introduction-Personalization

Personalized Federated Learning

- **Personalized federated learning**: customize personalized model for each client boosted by federated learning.
- **Categories**: Federated Transfer Learning, Federated Meta Learning, Federated Multi-Task Learning and Federated Distillation.
- **Federated Meta Learning**: each user's model is trained as a separate task; the shared model adapts to the **distribution of the tasks** instead of to each individual task

2

Meta-HAR Method

Meta-HAR Framework

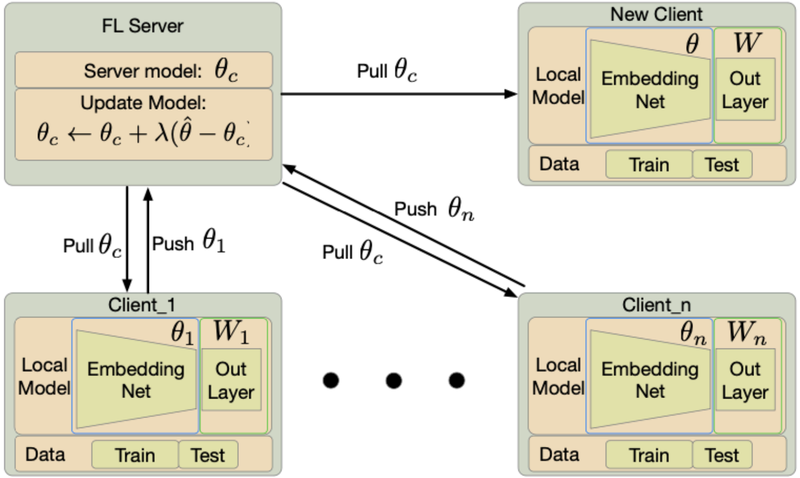

Fig. 2. An overview of the Meta-HAR framework.

- **Components**: One server (aggregating local updates), n clients (training on its private dataset).
- **Goal of the server**: updating a server model and applying it to a new client.
- **Goal of the clients**: training a personalized model.
- The client's model = embedding net + personalized classifier

Difference between Federated Learning:

- Only the embedding network is learned in a federated manner.
- Clients could have different classifiers.
- Clients' private data can have a different output activity set and even a different number of activity types

Local Model-Embedding Net

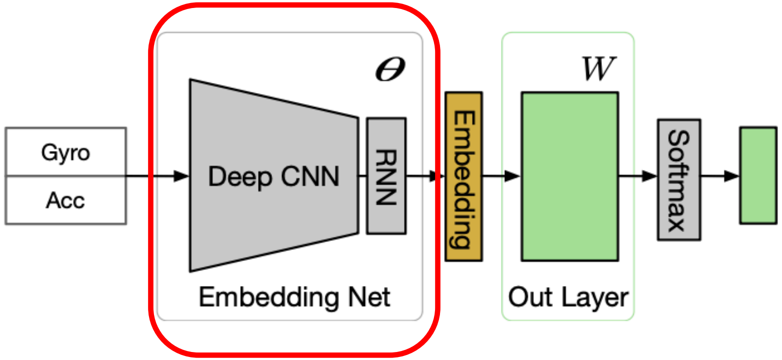

Fig. 3. Deep classifier structure for HAR.

- **Goal:** embed any given input signal, regardless of its activity type, into a fixed length vector.
- **Architecture :** Deep CNN + RNN
- **Training process**: a weight-sharing siamese network is used to predict whether two signal samples are of the same class or not.
- **Loss function selection**: pair-wise loss

Local Model-Embedding Net

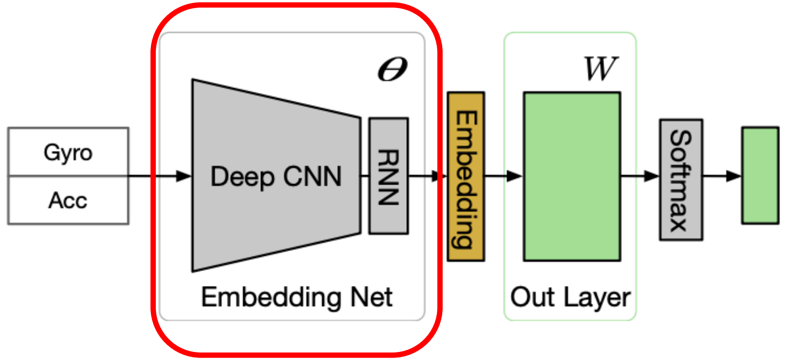

Fig. 3. Deep classifier structure for HAR.

Pair-wise Loss

Given input samples {(ei , ai), (ej, aj)}, where ei is embedding vector, ai is the label.

First, calculate the **cosine distance** of two vectors:

$$\varphi_{ij} = \frac{e_i^T e_j}{|e_i| \cdot |e_j|}$$

Then, compute **the pair-wise loss** based on the cosine distance:

$$l_{i,j} = -\delta(a_i, a_j)log(\sigma(\varphi_{ij})) - (1 - \delta(a_i, a_j))log(1 - \sigma(\varphi_{ij}))$$

where $\sigma(x)$ is sigmoid function;

$\delta(a_i, a_j) = 1 \text{ if } a_i = a_j, \text{otherwise } \delta(a_i, a_j) = 0$

It encourages the clustering of sample embeddings.

Local Model-Personalization

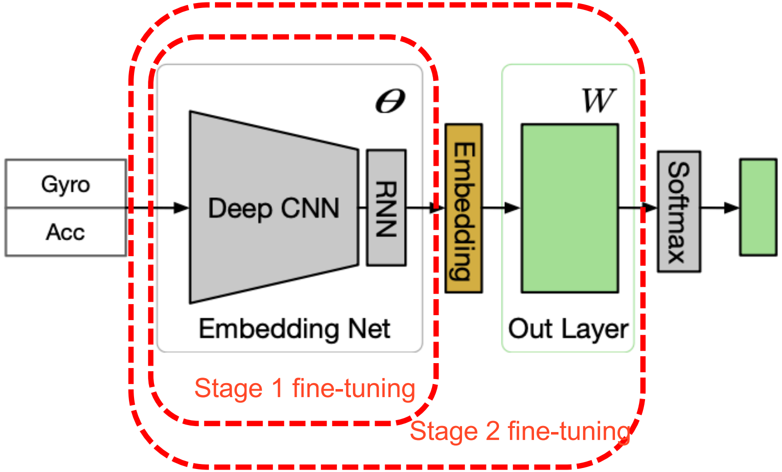

Fig. 4. Two-stage fine-tuning to get the personalized model.

Personalization

1. Initialize the embedding net with the server model.
2. Stage 1 fine-tuning: fine-tune embedding net with pair-wise loss on private data.
3. Stage 2 fine-tuning: jointly fine-tune the output layer parameters together with the embedding network with the cross-entropy loss.

- **Flexibility of personalized classifier**: number of classes is determined by private dataset

3 Experiments

Dataset Collection

<table>
<tr><th>Dataset</th><th>Users</th><th>Activities</th><th>Raw features</th><th>Scenario</th></tr>
<tr><td>Heterogeneous Human activity recognition (HHAR) Dataset [2]</td><td>9</td><td>6: {Standing, Sitting, Walking, Upstairs, Downstairs and Biking}</td><td rowspan="4">Acc_{x,y,z}, Gyro_{x,y,z}</td><td rowspan="3">Controlled environment</td></tr>
<tr><td>USC-HAD Dataset [3]</td><td>14</td><td>6: {Standing, Sitting, Walking, Upstairs, Downstairs and Running }</td></tr>
<tr><td>Merged Dataset</td><td>23</td><td>7: {Standing, Sitting, Walking, Upstairs, Downstairs and Running, Biking}</td></tr>
<tr><td>Collected Dataset</td><td>48</td><td>6: {Walking, Biking, (walking) Upstairs, (walking) Downstairs, Running and Taking Bus/Taxi }.</td><td>No constraints</td></tr>
</table>

[2]. Stisen, Allan, et al. "Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition." Proceedings of the 13th ACM conference on embedded networked sensor systems. 2015.
[3]. Zhang, Mi, and Alexander A. Sawchuk. "USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors." Proceedings of the 2012 ACM conference on ubiquitous computing. 2012.

Data Processing & split

1. Add an additional input dimension

compute the amplitude for each sensor. For gyroscope, the amplitude of sgx, sgy, sgz is

$$\sqrt{s_{gx}^2 + s_{gy}^2 + s_{gz}^2}$$

2. Split time-series signals into windows

split the sensor data into time windows, each window contains k readings;

k = 150;

3. Extract frequency components

apply a Fourier transformation to each axis of each segmented data block

4. Form input array

stack all the outputs of Fourier transformation, magnitudes and their corresponding frequencies, into a tensor of shape

$k \times 2(3 + 1) \times f$

5. Create Non-i.i.d distribution

for each user, randomly remove 0 to 2 activities from its local dataset

6. Split train-users and test-users

Meta-train users: participate in the meta-learning process

Meta-test users: their data are used to test the generalization ability of the meta-learned model.

Experiment Settings

- Num of meta-test users: HHAR or USC-HAR (1), Merged Dataset(1H1U), Collected Dataset (5)
- Five experimental models: Central, FedAvg, FedReptile[4], Meta-HAR, Meta-HAR-CE*
- Metric: Averaged test Accuracy
- Three fine-tuning methods:

    Separated: fine-tune embedding net, then fix embedding net and fine-tune classifier.

    Merged: jointly fine-tune embedding net and classifier.

    Two-stage: fine-tune embedding net, then jointly fine-tune both.

** Meta-HAR-CE: loss function of embedding net uses cross-entropy instead of pair-wise loss*

[4]. Jiang, Yihan, et al. "Improving federated learning personalization via model agnostic meta learning." arXiv preprint arXiv:1909.12488 (2019).

Results-Three Datasets

Algorithms	HHAR Dataset		USC-HAD Dataset		Collected Dataset	
	Meta-train user	Meta-test user	Meta-train user	Meta-test user	Meta-train user	Meta-test user
Central	98.55 ± 0.11	83.14 ± 8.40	99.31 ± 0.14	81.63 ± 10.51	90.18 ± 0.14	79.84 ± 0.41
FedAvg	79.56 ± 0.62	66.79 ± 1.84	84.44 ± 0.26	80.24 ± 1.54	69.19 ± 5.02	64.29 ± 6.86
FedReptile(1)	87.16 ± 0.29	86.00 ± 1.66	87.96 ± 0.21	85.33 ± 2.20	88.29 ± 0.66	83.96 ± 2.31
FedReptile(2)	92.64 ± 0.26	88.04 ± 2.65	91.02 ± 0.30	86.44 ± 3.05	90.84 ± 0.18	89.59 ± 1.29
FedReptile(3)	95.70 ± 0.25	91.84 ± 1.85	$\mathbf{93.98 \pm 0.31}$	89.17 ± 2.26	$\mathbf{91.49 \pm 0.31}$	92.30 ± 1.52
Meta-HAR(1)	98.32 ± 0.06	85.23 ± 1.80	92.01 ± 0.16	83.59 ± 2.28	89.16 ± 0.76	92.38 ± 0.43
Meta-HAR(2)	98.36 ± 0.04	91.25 ± 1.82	92.54 ± 0.11	90.19 ± 2.81	90.24 ± 0.63	$\mathbf{93.33 \pm 0.80}$
Meta-HAR(3)	$\mathbf{98.39 \pm 0.02}$	$\mathbf{92.50 \pm 1.26}$	93.79 ± 0.14	$\mathbf{91.07 \pm 1.74}$	90.76 ± 0.67	93.29 ± 1.03

Fig. 5. Test Results on HHAR, USC-HAD and collected Datasets. The number in the parenthesis denotes the fine-tuning epochs. All numbers are in percentage (%).

Results-Three Datasets

- A great performance degradation from Central to FedAvg.
- Meta-HAR and FedReptile achieved close performance.
- Meta-HAR and FedReptile outperform the Central on meta-test users.
- As the fine-tuning epochs increase, the acc is steadily improved.
- Due to the small size of local dataset, personalized models perform worst than Central on meta-train users on USC-HAD dataset.

Results-Merged Dataset

Algorithms	Meta-train	Meta-test (H)	Meta-test (U)
FedAvg	48.97 ± 0.62	39.99 ± 1.56	52.71 ± 1.97
FedReptile(1)	58.12 ± 0.55	59.52 ± 1.87	66.16 ± 2.52
FedReptile(2)	65.83 ± 0.65	66.35 ± 3.67	71.95 ± 2.72
FedReptile(3)	70.65 ± 0.53	69.40 ± 2.64	74.83 ± 2.19
Meta-HAR-CE(1)	94.05 ± 0.41	62.69 ± 1.90	61.37 ± 3.62
Meta-HAR-CE(2)	97.01 ± 0.36	62.74 ± 3.00	71.12 ± 2.52
Meta-HAR-CE(3)	**97.70 ± 0.24**	67.19 ± 3.47	80.96 ± 2.95
Meta-HAR(1)	91.42 ± 0.49	47.28 ± 1.65	73.98 ± 2.68
Meta-HAR(2)	93.64 ± 0.39	54.38 ± 2.47	87.00 ± 2.88
Meta-HAR(3)	95.35 ± 0.28	**75.83 ± 2.27**	**90.23 ± 2.44**

Fig. 6. Test Results on Merged Dataset. The number in the parenthesis denotes the fine-tuning epochs. All numbers are in percentage (%).

Results-Merged Dataset

- On meta-train users, the Meta-HAR model achieves an accuracy of 95.35% greatly outperform FedReptile which gives an accuracy of 70.65%.
- On meta-test users, Meta-HAR outperforms FedReptile on all datasets
- The higher performance achieved by Meta-HAR over Meta-HAR-CE on the meta-test user demonstrates the superior generalization ability of Meta-HAR.

Results-Fine-tuning Methods

Tune methods	Meta-train	Meta-test (H)	Meta-test (U)
Merged(1)	89.03 ± 0.83	47.60 ± 2.07	60.76 ± 3.62
Merged(2)	91.84 ± 0.62	48.47 ± 2.36	69.79 ± 2.62
Merged(3)	**95.38 ± 0.47**	50.38 ± 1.59	80.19 ± 3.39
Separated(1)	88.75 ± 0.65	47.22 ± 2.54	51.98 ± 1.79
Separated(2)	90.47 ± 0.69	48.69 ± 2.85	64.44 ± 1.16
Separated(3)	91.91 ± 0.71	50.98 ± 2.97	73.35 ± 1.72
Two-stage(3)	95.35 ± 0.28	**75.83 ± 2.27**	**90.23 ± 2.44**

Fig. 7. Test Results of Meta-HAR with different fine-tune methods on Merged Dataset. The number in the parenthesis denotes the fine-tuning epochs performed. All numbers are in percentage (%).

- Two-stage fine-tuning method significantly outperforms other approaches on the meta-test users.

4 Conclusions

Conclusions

- The heterogeneity in signal distribution and activity type distribution across users is widely existing in the HAR dataset.
- Personalized models can significantly outperform FedAvg.
- Federated Meta Learning shows better generalization ability than FedAvg.

References

[1]. Li, Chenglin, et al. "Meta-HAR: Federated Representation Learning for Human Activity Recognition." Proceedings of the Web Conference 2021. 2021.

[2]. Stisen, Allan, et al. "Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition." Proceedings of the 13th ACM conference on embedded networked sensor systems. 2015.

[3]. Zhang, Mi, and Alexander A. Sawchuk. "USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors." Proceedings of the 2012 ACM conference on ubiquitous computing. 2012.

[4]. Jiang, Yihan, et al. "Improving federated learning personalization via model agnostic meta learning." arXiv preprint arXiv:1909.12488 (2019).

Thanks