FedUSC: Collaborative Unsupervised Representation Learning from Decentralized Data for Internet of Things[1]

Jinglong Shen[1]

[1]Xidian University.

May 2023

[1]C. Zhao, Z. Gao, Y. Yang, *et al.*, "FedUSC: Collaborative Unsupervised Representation Learning from Decentralized Data for Internet of Things," *IEEE Internet of Things Journal*, 2023.

Table of Contents

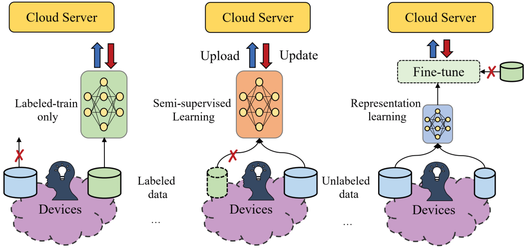
1 Background

2 Methodology

3 Experiments

4 Conclusion

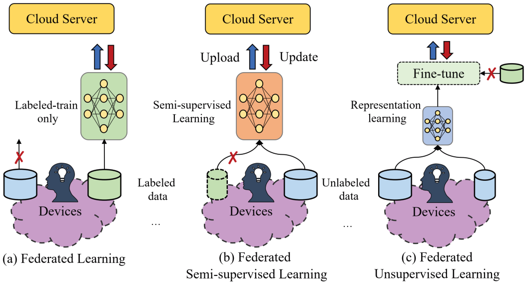
Federated Learning



(a) Federated Learning
(b) Federated Semi-supervised Learning
(c) Federated Unsupervised Learning

- **Advantages**: Better privacy performance. Lower communication overhead. Distributed training capabilities.
- **Limitations**: Still limited to supervised settings. While data on IoT devices usually come with few accompanying labels in real-world applications.

Federated Learning



- **Solutions**: Incorporating semi-supervised or unsupervised techniques into the federated learning frame work.

Table of Contents

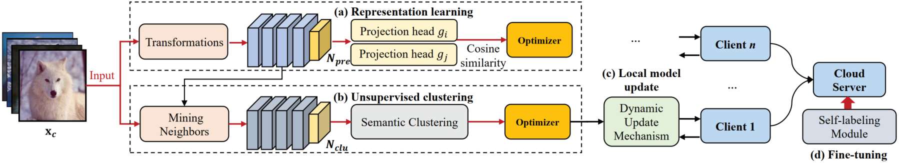
1 Background

2 Methodology

3 Experiments

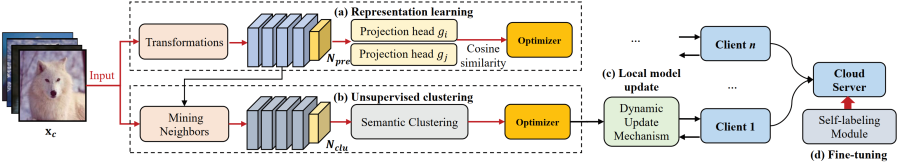
4 Conclusion

FedUSC Overview



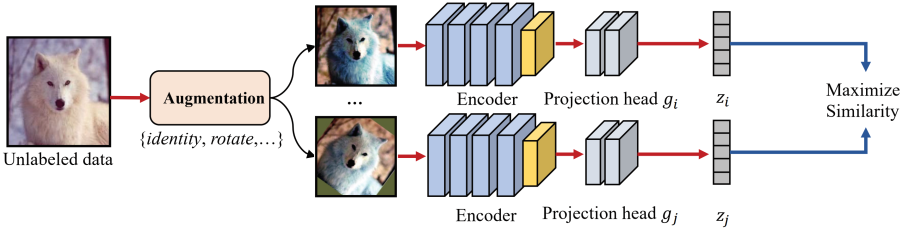
- **Representation learning**: each client conducts a pretext task through representation learning that can be used to obtain semantically meaningful features.
- **Semantic clustering**: clients can leverage prior knowledge to acquire image representation features, which can then be used to classify examples based on how similar the features are.

FedUSC Overview



- **Dynamic update mechanism**: the server aggregate weights and each client dynamically updates the local model according to weight divergence.
- **Self labeling**: the server mitigates the problem of noise inherent in the clustering process through the self-labeling method.

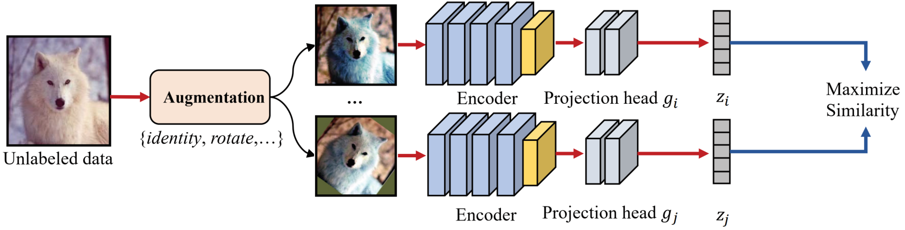
Representation Learning



RandAugment

- Several commonly used augmentations are considered, including crop, saturation, gray, contrast, hue, color, brightness, and stochastically choosing partial transformations to apply each augmentation.
- The distance between image samples $x \in \mathbf{x}_c$ and their augmentations $T[x]$ are going to be minimized. $\to \min d(f(x), f(T(x)))$

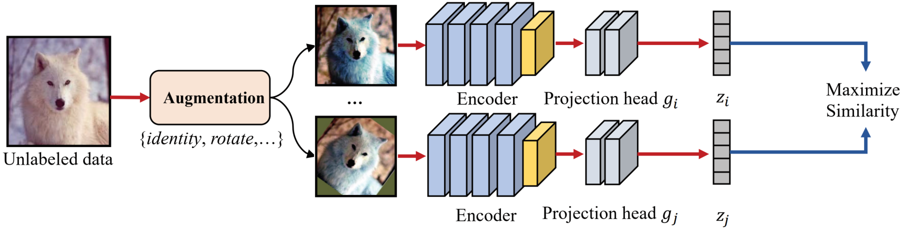
Representation Learning



Contrastive learning

- Two views are created using RandAugment and encoded via encoder (ResNet) to generate representations $h = f(x)$.
- The projection head $g(\cdot)$ is employed to map representations to the space where contrastive loss is applied. $\rightarrow z = g(f(x))$

Representation Learning

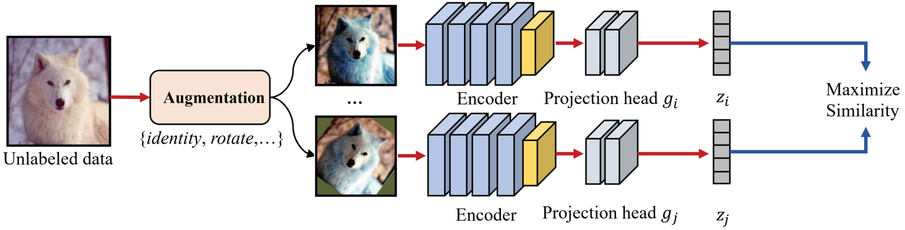


Contrastive learning

The normalized temperature-scaled cross-entropy loss is adopted to capture the distance between a positive pair of examples (i, j)

$$\mathcal{L}_{\text{pre}}^{c}(i,j) = -\log \frac{\exp\left(\text{sim}\left(z_i, z_j\right)/\tau\right)}{\sum_{m=1}^{2M} \mathbf{1}_{[m \neq i]} \exp\left(\text{sim}\left(z_i, z_m\right)/\tau\right)} \tag{1}$$

Representation Learning



Contrastive learning

$$\mathcal{L}_{\text{pre}}^{c}(i,j) = -\log \frac{\exp\left(\text{sim}\left(z_i, z_j\right)/\tau\right)}{\sum_{m=1}^{2M} \mathbf{1}_{[m\neq i]} \exp\left(\text{sim}\left(z_i, z_m\right)/\tau\right)} \quad (2)$$

where τ is the temperature scalar, i, j is the augmented samples from the same image, $sim(z_i, z_j) = \frac{z_i^T z_j}{\|z_i\|\|z_j\|}$ is cosine similarity between two images, $\mathbf{1}_{[m\neq i]}$ is indicator function evaluating to 1 if $m \neq i$.

Unsupervised Clustering

The unsupervised clustering setup is aim to learn a semantic clustering model N_{clu} (parameterized by a neural network with weights q) that can group together x and its neighbors $\mathcal{N}_x$.

$$\mathcal{L}^c_{clu} = -\frac{1}{|\mathbf{x}^c|} \sum_{x \in \mathbf{x}^c} \sum_{\hat{x} \in \mathcal{N}_x} \log < q(x), q(\hat{x}) > + \lambda \sum_{h \in \{1, \cdots, H\}} p_h \log p_h,$$
$$where \ \ p_h = \frac{1}{|\mathbf{x}_c|} \sum_{x \in \mathbf{x}_c} q_h(x), \tag{3}$$

where $\hat{x}$ represent the neighbors of x, $< \cdot >$ denotes inner product.

- The first term imposes q to make consistent predictions for x and its neighbors.
- The second term is used to avoid assigning all examples into one cluster.

Dynamic Update Mechanism

Each client trains $\mathcal{N}_{pre}$ and $\mathcal{N}_{clu}$ with E epochs and then uploads model to the server. To mitigate the adverse effects of data non-IID, the Dynamic update Mechanism (DUM) is further proposed to dynamically update the local clustering model q_c.

$$q_c^t = \begin{cases} q_g^t, & if \|q_c^{t-1} - q_g^t\|_2^2 \leq \mu \\ \xi q_c^{t-1} + (1-\xi) q_g^t, & if \|q_c^{t-1} - q_g^t\|_2^2 > \mu \end{cases} \quad (4)$$

where $\xi, \mu \in [0, 1]$ are decay rate and update threshold respectively, q_c is the c-th clients clustering model parameters, and t is the communication rounds.

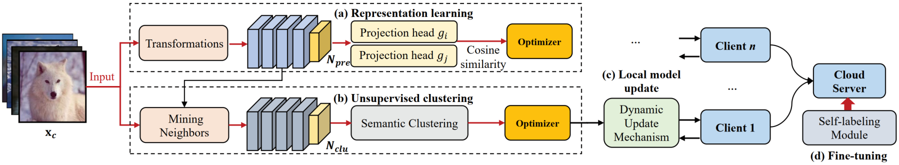
Self-Labeling Module

The self-labeling module utilizes fine-tuning capabilities to assign samples with highly confident predictions to the correct cluster.

$$\mathcal{L}_{self} = \frac{1}{|\mathbf{x}_g|} \sum_{x \in \mathbf{x}_g} \mathbf{1}(\max(q_g(x)) \geq \sigma) H(\hat{q}_g(x), q_g(x)), \tag{5}$$

where σ is the confidence threshold above which we retain pseudo-label, $q_g(x)$ is network softmax output and $\hat{q}_g(x) = \mathrm{argmax}(q_g(x))$ is pseudo-label of x, which we use argmax turn a probability distribution into a one-hot distribution. $H(\cdot)$ is the standard cross-entropy loss on pseudo-label.

Unsupervised Learning Problem Definition



Given a set of IoT clients $C = \{c_1, c_2, \cdots, c_n\}$ and a global server G, with local dataset $\mathbf{x}$ and public dataset $\mathbf{x}_g$, respectively. The total objective function $\mathcal{L}_\Phi$ can be represented as

$$\min \mathcal{L}_\Phi = \mathcal{L}_{\text{pre}}(\mathbf{x}) + \mathcal{L}_{\text{clu}}(\mathbf{x}) + \mathcal{L}_{\text{self}}(\mathbf{x}_g) \tag{6}$$

where pretext task loss $\mathcal{L}_{pre}$ and semantic clustering loss $\mathcal{L}_{clu}$ are used to learn model representation and mine nearest instances of the same semantic cluster respectively, and self labeling loss $\mathcal{L}_{self}$ is used to fine-tuning the shared model.

Table of Contents

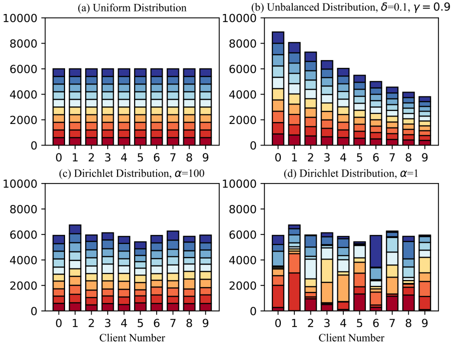
1 Background

2 Methodology

3 Experiments

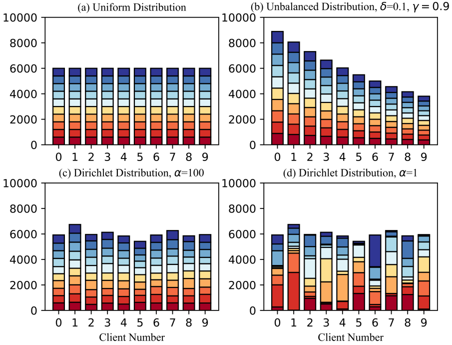
4 Conclusion

Experimental setup



- **Datasets**: CIFAR-10, SVHN, STL-10, Mini-ImageNet, and COVID-19.
- **Model**: The standard ResNet-18 backbone is used for representation learning and semantic clustering.

Experimental setup



- **Implementation details**: The default settings for all experiments are R = 200 communication rounds, E = 200 local training epochs, n = 10 clients with 4 randomly selected for local training in each round, and non-IID training with unbalanced distribution.

Ablation Studies

Augmentation	Acc. (%)
Crop & flip	65.16
Cutout	66.74
RandAugment	**68.35**

(a) Ablation of augmentation

Components	Acc. (%)
RotNet	57.39
Inst. discr.	65.53
SimCLR	**68.35**

(b) Ablation of pretext task

Update	Acc. (%)
FedAvg	58.73
FedProx	62.19
ASTW	57.44
FedEMA	68.08
DUM	**68.35**

(c) Ablation of update method

Methods	$\alpha = 100$	$\alpha = 1$
W/O RandAugment	67.92	60.44
W/O semantic clustering	67.26	58.48
W/O self-labeling	60.33	54.17
FedUSC (with default settings)	**69.04**	**61.51**

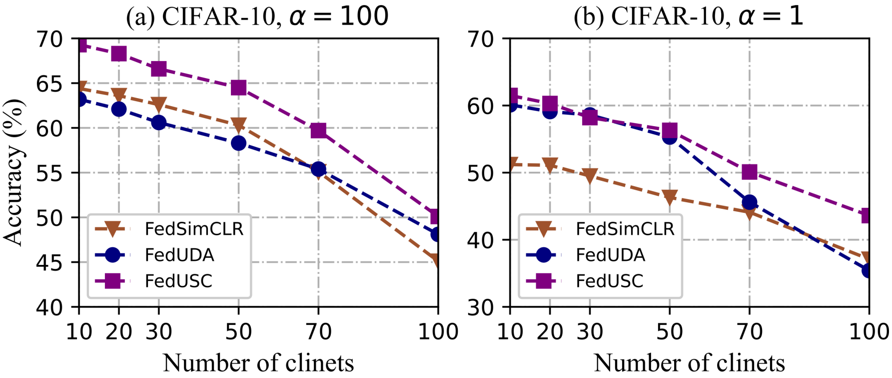
(d) Ablation study of our improvements

Comparison with the state-of-the-art

Methods	Architecture	Param.	Uniform	Unbalanced ($\delta = 0.1, \gamma = 0.9$)	Dirichlet ($\alpha = 100$)	Dirichlet ($\alpha = 1$)
Upper-bound methods						
FedAvg [1] (supervised)	ResNet-50	23M	91.53	-	-	-
SCAN [35] (centralized)	ResNet-50	23M	88.30	-	-	-
Federated Semi-supervised learning methods						
DS-FL [26]	VGG-16	93M	60.17	57.39	57.63	46.98
FedMatch (labels-at-server) [10]	ResNet-50	23M	66.42	60.74	62.37	57.85
Federated Unsupervised learning methods						
FedSimCLR [32]	ResNet-50	23M	64.37	60.41	61.03	51.22
FedU [20]	ResNet-50	23M	59.72	56.58	57.13	53.85
Our methods with different network architecture						
FedUSC	ResNet-18	11M	61.47	60.39	61.12	54.84
FedUSC	ResNet-50	23M	64.26	61.84	62.29	56.44
FedUSC (self-labeling)	ResNet-50	23M	**69.54**	**68.35**	**69.04**	**61.51**

Comparison with the state-of-the-art

Performance under different number of clients



Comparison with the state-of-the-art

Performance under different datasets

Methods	SVHN	STL-10	Mini-ImageNet	COVID-19
Uniform				
FedSimCLR [32]	65.47	58.54	60.37	74.37
FedMatch [10]	64.89	58.32	58.31	73.45
FedUSC	68.71	58.44	63.94	77.75
Unbalanced				
FedSimCLR [32]	60.52	56.47	56.73	72.26
FedMatch [10]	61.38	55.41	57.03	70.33
FedUSC	66.10	57.86	61.08	75.37
Dirichlet				
FedSimCLR [32]	53.13	52.73	48.53	61.63
FedMatch [10]	55.47	50.36	47.52	65.66
FedUSC	61.57	57.88	56.94	71.13

Table of Contents

1 Background

2 Methodology

3 Experiments

4 Conclusion

Conclusion

Limitations

- The the number of clusters is a hyperparameter that need to be specified manually.
- Additional computation overhead is introduced to train auxiliary networks.

Thank You!