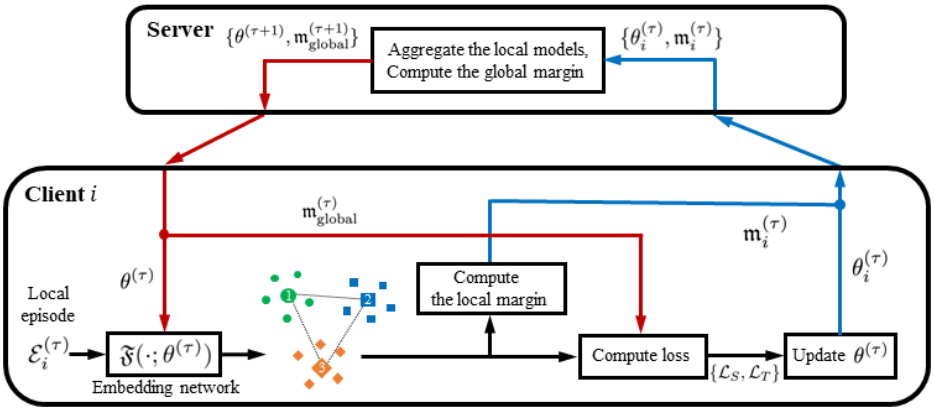
MetaVers: Meta-Learned Versatile Representations with Large Margins for Personalized Federated Learning



Installation

- Create a virtual environment with `conda create`
  - `conda create -n mvs python=3.8.5`
  - `conda activate mvs`
  - `pip install torch==1.7.1+cu110 torchvision==0.8.2+cu110 torchaudio==0.7.2 -f https://download.pytorch.org/whl/torch_stable.html`
  - `pip install learn2learn`
  - `pip install tensorboardX`
  - `pip install tensorflow`
  - `pip install sklearn`

Fast Run

Results on CIFAR-10

- Run: `python trainer.py --dataset cifar10 --rand 0 --seed_list 42`

Results on CIFAR-100

- Run: `python trainer.py --dataset cifar100 --rand 0 --seed_list 42`

Results on CINIC-10

- Run: `python trainer.py --dataset cinic --rand 0 --seed_list 42`

For Details

`--loss`: What loss to use, for cross-entropy; `ce` or tirplet; `riplet` . Default: `hybrid`

`--d_from`: How to set the margin of the parameter used for triplet loss, for fixed margin;`margin` Default: `select`

`--w`: Interval Value W. Default: `50`

`--control`: Non-random way, according to the benchmark maximum way. Default: `fixed`

`--seed_list`: Receive a list of how to make a random seed (adjust the result randomness)

`--method`: Whether episodes are created locally in a decentralized setting, for centralized setting; `centralized` Default: `decentralized`

`--lr`: Learning rate for gradient update. Default: `0.001`

`--optimizer`: Whether to use decay for Adam optimizer. When using a weight decay of 1e-3: `1` Default: `0`

`--evaluation_unit`: Evaluate using the validation set every n round. Default: `50`

`--activated`: Number of the activated clients. Default: `5`

`--embedder`: which model to use.[`conv4`, `lenet`,`resnet18`] Default: `lenet`

`--gamma`: loss balancing hyper parameter. Default: `0.5`

`--rand`: Fix the randomness of the PyTorch according to the seed.(adjust the result randomness) Default: `1`

MetaVers: PFL Results Performance Reproduction

Table 1. Test accuracy (± SEM) on CIFAR-10, CIFAR-100, and CINIC-10.

	CIFAR-10			CIFAR-100			CINIC-10		
# clients # samples/client	50 800	100 400	500 80	50 800	100 400	500 80	50 1800	100 900	500 180
Local	84.8 ± 0.1	82.1 ± 0.2	76.2 ± 0.2	49.8 ± 0.2	44.5 ± 0.4	31.0 ± 0.3	58.4 ± 0.2	57.4 ± 0.4	50.3 ± 0.0
FedAvg [12]	56.4 ± 0.5	59.7 ± 0.5	54.0 ± 0.5	23.6 ± 0.2	24.0 ± 0.2	20.4 ± 0.0	45.6 ± 0.4	44.7 ± 0.5	45.7 ± 0.5
LG-FedAvg [9]	87.9 ± 0.3	83.6 ± 0.7	64.7 ± 0.7	43.6 ± 0.2	37.5 ± 0.9	20.3 ± 0.5	59.5 ± 1.1	59.9 ± 2.1	52.5 ± 0.8
pFedMe [4]	86.4 ± 0.8	85.0 ± 0.3	80.3 ± 0.5	49.8 ± 0.5	47.7 ± 0.4	32.5 ± 0.8	69.9 ± 0.5	68.9 ± 0.7	58.8 ± 0.1
FedU [5]	80.6 ± 0.3	78.1 ± 0.5	65.6 ± 0.4	41.1 ± 0.2	36.0 ± 0.2	15.9 ± 0.4	59.3 ± 0.2	55.4 ± 0.6	41.6 ± 0.5
FedProto [16]	85.9 ± 0.7	79.0 ± 0.4	51.0 ± 0.0	47.8 ± 0.5	17.8 ± 0.1	10.9 ± 0.1	58.2 ± 0.7	40.3 ± 0.8	26.0 ± 0.0
Per-FedAvg [2]	71.1 ± 1.5	79.1 ± 3.7	67.7 ± 1.9	38.2 ± 2.0	34.1 ± 0.4	32.8 ± 1.7	53.8 ± 0.8	53.5 ± 0.6	59.6 ± 0.7
pFedHN [15]	90.2 ± 0.6	87.4 ± 0.2	83.2 ± 0.8	60.0 ± 1.0	52.3 ± 0.5	34.1 ± 0.1	70.4 ± 0.4	69.4 ± 0.5	64.2 ± 0.1
pFedGP [1]	89.2 ± 0.3	88.8 ± 0.2	87.6 ± 0.4	63.3 ± 0.1	61.3 ± 0.2	50.6 ± 0.2	71.8 ± 0.3	71.3 ± 0.4	68.1 ± 0.3
FedPer [2]	83.8 ± 0.8	81.5 ± 0.5	76.8 ± 1.2	48.3 ± 0.6	43.6 ± 0.2	25.6 ± 0.3	70.6 ± 0.2	68.4 ± 0.5	62.2 ± 0.1
FedRep [3]	82.4 ± 1.5	80.7 ± 1.0	77.3 ± 0.8	45.1 ± 2.8	38.8 ± 1.1	30.2 ± 0.4	67.1 ± 1.1	64.7 ± 0.0	61.5 ± 0.5
kNN-Per [11]	89.6 ± 0.6	89.5 ± 0.4	84.8 ± 0.4	61.8 ± 0.3	56.0 ± 0.3	38.7 ± 0.7	71.8 ± 0.2	72.0 ± 0.2	69.2 ± 0.6
FedBABU [13]	58.9 ± 1.0	58.5 ± 0.6	57.9 ± 0.5	22.7 ± 0.4	21.8 ± 0.7	17.0 ± 0.6	49.9 ± 0.4	50.7 ± 0.2	50.9 ± 1.3
Ours									
(K, Q)	$(20, 30)$	$(20, 30)$	$(20, 30)$	$(10, 20)$	$(10, 20)$	$(3, R^*)$	$(25, 35)$	$(25, 35)$	$(20, 30)$
`MetaVers` (only $\mathcal{L}_T$)	90.3 ± 0.2	89.7 ± 0.2	89.6 ± 0.1	64.7 ± 0.2	62.9 ± 0.3	46.6 ± 0.2	72.6 ± 0.4	72.7 ± 0.2	71.9 ± 0.2
`MetaVers` (only $\mathcal{L}_S$)	89.9 ± 0.3	88.6 ± 0.1	88.1 ± 0.1	$\mathbf{66.7 \pm 0.1}$	64.6 ± 0.2	54.4 ± 0.3	72.8 ± 0.3	72.5 ± 0.2	71.7 ± 0.1
`MetaVers`	$\mathbf{90.8 \pm 0.3}$	$\mathbf{90.2 \pm 0.2}$	$\mathbf{89.9 \pm 0.3}$	$\mathbf{66.7 \pm 0.0}$	$\mathbf{64.8 \pm 0.1}$	$\mathbf{55.8 \pm 0.1}$	$\mathbf{73.2 \pm 0.4}$	$\mathbf{73.2 \pm 0.3}$	$\mathbf{72.5 \pm 0.1}$

Samples/client indicates the mean number of samples per local client in each case.
(K, Q) indicates the number of samples per class corresponding to the support set and the query set constituting the training episode.
R^* indicates the remaining data samples per class after securing K samples by prioritizing the support set configuration.

PFL Results Performance Reproduction on CIFAR-10

- `python trainer.py --exp_name reproduction --loss hybrid --d_from select --w 50 --control fixed --seed_list 0 21 42 --dataset cifar10 --num_user 50 --optimizer 1 --train_shot 20 --train_query 30 --gamma 0.4`
- `python trainer.py --exp_name reproduction --loss hybrid --d_from select --w 50 --control fixed --seed_list 0 21 42 --dataset cifar10 --num_user 100 --optimizer 1 --train_shot 20 --train_query 30 --gamma 0.3`
- `` `python trainer.py --exp_name reproduction --loss hybrid --d_from select --w 50 --control fixed --seed_list 0 21 42 --dataset cifar10 --num_user 500 --optimizer 1 --train_shot 20 --train_query 30 --gamma 0.2 ``

PFL Results Performance Reproduction on CIFAR-100

- `python trainer.py --exp_name reproduction --loss hybrid --d_from select --w 50 --control fixed --seed_list 0 21 42 --dataset cifar100 --num_user 50 --optimizer 0 --shot_500 3 --train_shot 10 --train_query 20 --gamma 0.6`
- `python trainer.py --exp_name reproduction --loss hybrid --d_from select --w 50 --control fixed --seed_list 0 21 42 --dataset cifar100 --num_user 100 --optimizer 0 --shot_500 3 --train_shot 10 --train_query 20 --gamma 0.6`
- `python trainer.py --exp_name reproduction --loss hybrid --d_from select --w 500 --control fixed --seed_list 0 21 42 --dataset cifar100 --num_user 50 --optimizer 0 --shot_500 3 --train_query 15 --gamma 0.6`

PFL Results Performance Reproduction on CINIC-10

- `python trainer.py --exp_name reproduction --loss hybrid --d_from select --w 50 --control fixed --seed_list 0 21 42 --dataset cinic --num_user 50 --optimizer 0 --train_shot 25 --train_query 35 --gamma 0.5`
- `python trainer.py --exp_name reproduction --loss hybrid --d_from select --w 50 --control fixed --seed_list 0 21 42 --dataset cinic --num_user 100 --optimizer 0 --train_shot 25 --train_query 35 --gamma 0.5`

- `python trainer.py --exp_name reproduction --loss hybrid --d_from select --w 50 --control fixed --seed_list 0 21 42 --dataset cinic --num_user 500 --optimizer 0 --train_shot 20 --train_query 30 --gamma 0.5`

Additional Experiment

[1] Only one Loss Example

- **Triplet Loss** (only $\mathcal{L}_T$)
  `python trainer.py --loss triplet --d_from margin --margin 3.0 --w 50 --dataset cinic --num_user 500 --exp_name triplet --seed_list 0 21 42 --optimizer 0 --train_shot 25 --train_query 35 --gamma 0.5 --lr 0.001 --gpu_number 0`
- **Cross-entropy Loss** (only $\mathcal{L}_S$)
  `python trainer.py --loss ce  --dataset cinic --num_user 500 --exp_name ce --seed_list 0 21 42 --optimizer 0 --train_shot 20 --train_query 30 --gamma 0.5 --lr 0.001 --gpu_number 0`

[2] Lower Way Example

- `python trainer.py --loss hybrid --d_from select --w 50 --dataset cifar100 --num_user 100   --exp_name table4_low --low_way 5 --seed_list 0 21 42 --optimizer 0 --train_shot 10 --train_query 20`
- **Random Way**
  `python trainer.py --random_way various --loss hybrid --d_from select --w 50 --dataset cinic --num_user 100   --exp_name table4_random --low_way 3 --seed_list 0 21 42 --optimizer 0 --train_shot 25 --train_query 35`

[3] Centralized Learning Example

- `python trainer.py --version eccv --loss hybrid --d_from select --w 50 --dataset cifar100 --num_user 50 --exp_name table5_central --method centralized --seed_list 0 21 42 --optimizer 0 --train_shot 10 --train_query 20`

CINIC-10 Dataset Download

- If the CINIC-10 dataset is not downloaded correctly, you can download it directly from the link and put it in the "'./cinic" folder.
  https://datashare.is.ed.ac.uk/bitstream/handle/10283/3192/CINIC-10.tar.gz