AML&XAI

Assignment 2 : Optimization-based Meta-Learning

1. Goal

Solve the few-shot learning problem via optimization-based meta-learning

2. Development environment

- python (>=3.6) on Linux
- pytorch (>=1.6)

3. Template file

Uncompleted code zip 1

- Execution using main.py
    - Example of execution command

      python3 main.py --trainer maml --dataset omniglot --n_way 5 --k_spt 5 --k_qry 10 --task_num 16 –inner_lr 0.4 --meta_lr 0.001 --inner_step 5 --inner_step_test 10

      python3 main.py –trainer reptile --dataset omniglot --n_way 5 --k_spt 5 --k_qry 10 --task_num 16 –inner_lr 0.4 --meta_lr 0.001 --inner_step 5 --inner_step_test 10 **--batch-size 5**

      The above commands are just examples. You should search the hyperparameters for optimization (meta_lr, inner_lr, task_num, inner_step, k_qry) proper to each dataset.
- Trainer folder : There are three files including the following methods
    - reptile.py (Reptile: On First-Order Meta-Learning ALgorithms)
    - maml.py (MAML & FOMAML: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks)
- Data_handler folder : Pre-defined loader that produces few-shot learning tasks by processing whole datasets (Omniglot, Sine wave)
- Network folder : There exists two kinds of neural network architectures
    - Sine wave dataset is trained with 2 layer MLP network
    - Onniglot dataset is trained with 4 layer convolution network

4. Experiment details

- MAML & FOMAML
    - Inner loop optimizer: SGD (Please use **torch.autograd.grad** to compute the gradients of adapted weights, and apply **SGD without momentum and weight decay manually**. The details on the input arguments of torch.autograd.grad are shown in pytorch documentation. For implementing the inner loop in MAML, we recommend using the **manually designed SGD** instead of using the optimizers in torch.optim)
    - Outer loop optimizer: ADAM
- Reptile
    - Inner loop optimizer: ADAM

- For training a meta learner, it is important to **carefully tune the learning rates** on inner-loop (inner_lr) and outer loop (meta_lr) to improve the performance of a learner.

5. Few-shot learning scenario

- Sine wave : 5-shot or 10-shot regression task
  - $y = a * \sin(x + b * \pi)\,, a \sim U(0.1,5.0), b \sim U(0,1), x \sim U(-5,5)$
  - a: Amplitude, b: Coefficient for phase, x: Randomly selected points
- Omniglot : 5-way 1-shot or 5-shot classification task

6. To Do

- Complete two script files for each method (Reptile, MAML & FOMAML) in Trainer folder
- For the Sine wave dataset, train meta learners using each method on 5-shot and 10-shot regression tasks and evaluate the learner on 5-shot and 10-shot regression task, respectively. For reporting the results, submit the following plots

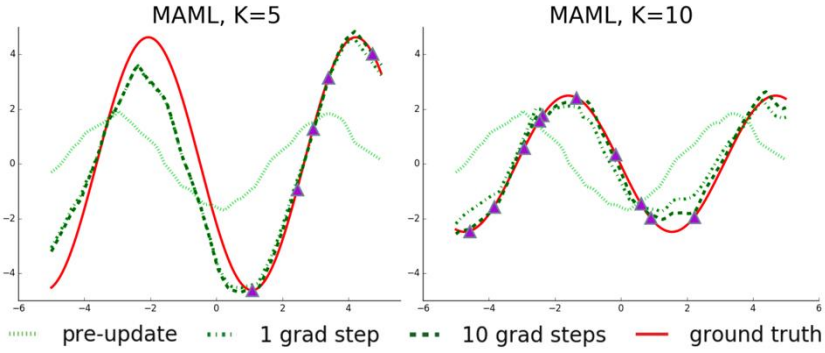

Figure 1 Results for few-shot regression task.
The purple triangle points are data used for adaptation at meta-test phase

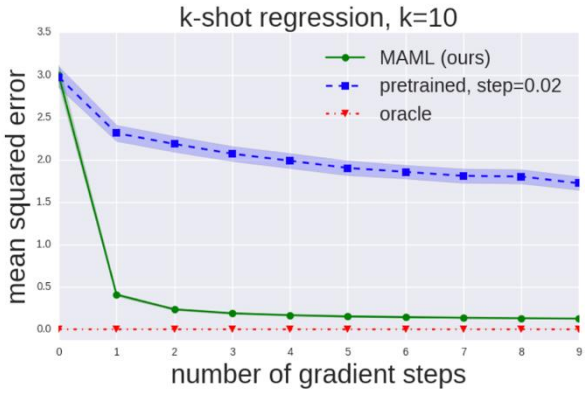

Figure 2 Results for fast adaptation on few-shot regression task.

  - Figure 2 shows how quickly MAML can adapt to a novel task. You should submit two plots for Figure 2 including the results for 5-shot and 10-shot regression task. You don't have to report the results for pretrained and oracle.
  - You should use newly sampled amplitude, phase, and input points at meta-test phase
- For the Omniglot dataset, train meta learners using FOMAML and Reptile on 1-shot and 5-shot classification tasks and evaluate the learner on 1-shot and 5-shot classification tasks, respectively. You don't have to consider MAML for the Omniglot dataset.

    - For reporting the results, submit the plots like Figure 2. You should submit two plots including the results for 1-shot and 5-shot classification task. **Accuracy of trained models used to plot the results should have more than 85%.**
- In HW2, you can report all the results **fixing the seed number as 0**.

7. Scoring rule

- Total 100 points
    - 20 points : MAML and FOMAML are implemented well
    - 20 points : Reptile is implemented well
    - 20 points : Figures for fast adaptation on Sine wave dataset
    - 20 points : Figures for fast adaptation on Omniglot dataset
    - 20 points : report (You can use any editor, e.g., word, latex etc.)

8. Submission due

- You have to submit two **source files** for methods (reptile.py, maml.py), **trained models** for each method and scenario (having acc > 85%) and the **report** until 2021. 5. 24. 11:59 pm.