Meta Learning Assignment 2

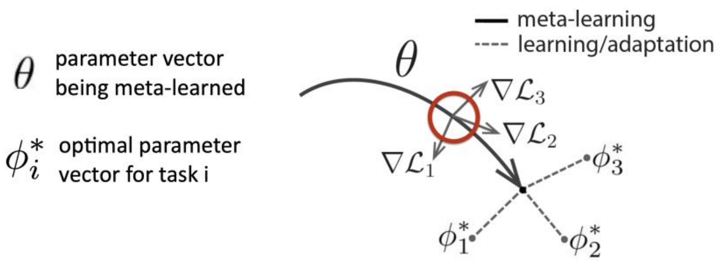
Instructor: Taesup Kim

Deadline: 2022/05/19 Thursday, 23:59

- Compress your report, code, and results into a single zip file and upload it to ETL.
    - File name: {Name}_{Student Number}_Assignment2.zip
    - Ex) 송재헌_2020-12345_Assignment.zip
- Be punctual! Note that 10%p of your score will be deducted for every hour you are late
- It is recommended to start early. Training the model can take a long time.
- Try to optimize your code the best you can. Unreasonably inefficient codes can result in deducted points.
- Explicitly mention any collaborators or code references (e.g., GitHub). Plagiarism will not be tolerated.

1 Model-Agnostic Meta-Learning (30 points)

Algorithm Overview



The main objective of MAML [1] is to (meta-)learn parameters θ that can quickly be adapted to a given task. We define the loss $\mathcal{L}$ of a model with parameters ϕ on the data $\mathcal{D}_i$ of a task $\mathcal{T}_i$ as:

$$\mathcal{L}(\phi, \mathcal{D}_i) = \frac{1}{|\mathcal{D}_i|} \sum_{(x^j, y^j) \in \mathcal{D}_i} -\log p_\phi\left(y = y^j \mid x^j\right)$$

The task-specific parameters ϕ are optimized during adaptation, often called the *inner loop*. For a task $\mathcal{T}_i$ and L inner loops, adaptation is carried out as follows:

$$\phi_i^1 = \theta - \eta \nabla_\theta \mathcal{L}\left(\theta, \mathcal{D}_i^{sup}\right)$$

$$\phi_i^2 = \theta - \eta \nabla_{\phi_i^1} \mathcal{L}\left(\phi_i^1, \mathcal{D}_i^{sup}\right)$$

$$\vdots$$

$$\phi_i^L = \theta - \eta \nabla_{\phi_i^{L-1}} \mathcal{L}\left(\phi_i^{L-1}, \mathcal{D}_i^{sup}\right)$$

Notice that only the support data is used to adapt the parameters to ϕ_i^L. To optimize θ in the *outer* loop, we use the same loss function applied to the adapted parameters and the query data. Note that the query data and support data both come from the same data distribution such that $\mathcal{D}_i^{sup}, \mathcal{D}_i^{query} \sim p(\mathcal{D}_i)$.

$$\theta_{t+1} \coloneqq \theta_t - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}\left(\phi_i^L, \mathcal{D}_i^{query}\right)$$

Here, a single MAML gradient $\mathcal{L}(\phi_i^L)$ is calculated following the chain rule such that

$$\nabla_\theta \mathcal{L}\left(\phi_i^L, \mathcal{D}_i^{query}\right) = \nabla_{\phi_i^L} \mathcal{L}\left(\phi_i^L, \mathcal{D}_i^{query}\right) \cdot \prod_{k=1}^{L} \left(I - \alpha \nabla_{\phi_i^{k-1}} \left(\nabla_\theta \mathcal{L}\left(\phi_i^{k-1}, \mathcal{D}_i^{query}\right)\right)\right)$$

For this assignment, we consider a **first-order variant of MAML** (FO-MAML) which reduces the bulk of the required computation by ignoring the second derivative marked in red. Thus, we can *approximate* the meta-gradients as follows.

$$\nabla_\theta \mathcal{L}\left(\phi_i^L, \mathcal{D}_i^{query}\right) \approx \nabla_{\phi_i^L} \mathcal{L}\left(\phi_i^L, \mathcal{D}_i^{query}\right)$$

Problems:

1. [20 points] In the `maml.py` file, complete the implementation of the `MAML.inner_loop` and `MAML.outer_loop` methods. The former computes the task-adapted network parameters (and accuracy metrics), and the latter computes the MAML objective. Pay attention to the inline comments.

   Hint: we can approximate the meta-gradients with Monte Carlo estimation on minibatches of data. Thus, for one minibatch of size B, we have:

   $$\nabla_\theta \mathcal{L}\left(\phi_i, \mathcal{D}_i^{query}\right) \approx \frac{1}{B} \sum_{k=1}^{B} \nabla_{\phi_i} \mathcal{L}\left(\phi_i, \mathcal{D}_i^k\right) \; s.t. \quad \cup_k \mathcal{D}^k = \mathcal{D}^{query}$$

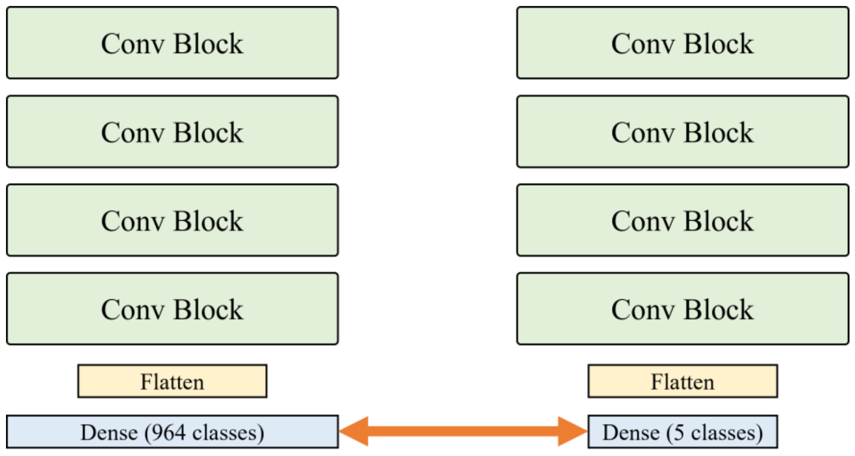
   Hint: the simplest way to implement the inner loop and outer loop involves using tf.GradientTape.

   Hint: Reading documentation on writing a training loop from scratch might help.

2. [10 points] Assess your implementation of FO-MAML on 5-way 5-shot Omniglot. Use 5 inner loop steps with a fixed learning rate of 0.4. Use 15 query examples per class per task. You should not need to adjust the outer learning rate from its default of 0.001.
   Submit: a plot of the validation post-adaptation query accuracy over the course of training. You should obtain a query accuracy on the validation split of at least 80% with 1000 iterations.
   Submit: *final* model weights and report all hyperparameters during training.
   Extra Credit (+10 points): MAML can exhibit very unstable training performance. To solve this problem [2] proposes MAML++, an alternative to MAML that shows remarkable stability during training. Modify `maml.py` to implement MAML++ and submit two validation plots comparing the training stability between the two methods.

2 Transfer Learning vs. Meta-Learning (20 points)

Transfer learning aims to improve the process of learning tasks using the experience gained by solving tasks which are somewhat similar to the target task. In this assignment, we will compare the effectiveness of transfer learning when compared to meta-learning.



Algorithm Overview

First, train a convolutional model defined in maml.py on the *entire* Omniglot train dataset. Next, freeze the convolutional layers, discard the classification layer, and replace it with a new classification layer that outputs 5 classes (for a 5-way 5-shot task). Adapt the new model on the support dataset using a few gradient steps and validate the performance of the model on the query dataset.

Problems:

1. [10 points] In the `maml_transfer.py` file, complete the implementation of the `transfer_learn` method. In this method, we create a new model using frozen feature extractors, adapt the model into sampled support tasks, and finally validate the model on query tasks. Pay attention to inline comments

2. [10 points] Assess the performance of transfer learning when compared to FO-MAML. Does the performance increase or decrease? Explain why.
   Submit: a plot of the validation post-adaptation query accuracy of the transfer-learned model over the course of training.
   Submit: final model weights of the *pretrained model* and report all hyperparameters during training.

References

[1] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126-1135, JMLR. Org,2017.

[2] Antreas Antoniou, Harrison Edwards and Amos Storkey. How to train your MAML. In *the sixth International Conference on Learning Representations*, 2018