Meta-Learner with Linear Nulling: Supplementary Material

Sung Whan Yoon
shyoon8@kaist.ac.kr

Jun Seo
tjwns0630@kaist.ac.kr

Jaekyun Moon
jmoon@kaist.edu

School of Electrical Engineering,
Korea Advanced Institute of Science and Technology (KAIST)

A Details of Learning and Classification Procedures

Algorithm 1 provides detailed steps of the initial learning procedure of our meta-learner. For each training episode, N_c classes are randomly chosen from the training set of a given dataset. Then, for each class, N_s labeled samples are randomly chosen as the support set S_k, and N_q labeled samples are chosen as the query set Q_k, without any overlapping samples between S_k and Q_k. With the support set S_k, the average network output vector $\bar{\mathbf{g}}_k$ is obtained for each class (in line 5). Based on the per-class average network output vectors, error vectors are obtained for all classes (in line 6) without any relabeling on the reference vectors. Then the linear transformer $\mathbf{M}$ is computed as a null-space of the error signals. For each query input, the Euclidean distances to the reference vectors in the projection space $\mathbf{M}$ are measured, and the training loss is computed using these distances. The average training loss is obtained over all N_q query inputs of N_c classes (in line 11 to 14). The learnable parameters θ of the embedding network and the references $\mathbf{\Phi}$ are now updated with the average training loss (in line 16).

B Hyperparameters in Experiment

In Table 1, we show the hyperparameters used for 20-way Omniglot and 5-way *mini*ImageNet experiments in the main paper. For all experiments, the initial learning rate is 10^{-3}, but the rate decays by half in every S_d episodes in the *mini*ImageNet experiments. S_d, the learning rate decay step, and N_q, the number of query images per class in each episode, are chosen empirically.

Table 1: Optimized hyperparameters for 20-way Omniglot and 5-way *mini*ImageNet experiments

Experiment	S_d	N_q
20-way Omniglot 1-shot	No decay	7
20-way Omniglot 5-shot	No decay	7
5-way *mini*ImageNet 1-shot	5000	5
5-way *mini*ImageNet 5-shot	7500	2

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

Algorithm 1 Initial learning is done by N_E training episodes. Each episode E_i consists of N (image, label) pairs. These N shots are composed of N_c classes and there are N_s shots and N_q queries in each class. L_{train} is the loss for training learnable parameters. The Euclidean distance between two vectors is denoted as $d(\cdot, \cdot)$.

Input: Training set $E^T = \{E_1, ..., E_{N_E}\}$ where $E_i = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ is an episode with $N = N_c(N_s + N_q)$ pairs of image and label where $y_n \in \{0, ..., N_c - 1\}$. $E_i^{(k)} = \left\{(\mathbf{x}_1^{(k)}, y_1^{(k)}), ..., (\mathbf{x}_{N_s+N_q}^{(k)}, y_{N_s+N_q}^{(k)})\right\}$ is the subset of E_i consisting of all pairs $(\mathbf{x}_n, y_n)$ such that $y_n = k$.

for i in $\{1, ..., N_E\}$ **do**
  $L_{train} \leftarrow 0$
  for k in $\{0, ..., N_c - 1\}$ **do**
    $S_k \leftarrow \left\{(\mathbf{x}_n^{(k)}, y_n^{(k)})\right\}$ with $(\mathbf{x}_n^{(k)}, y_n^{(k)}) \in E_i^{(k)}, n \leq N_s$
    $\bar{\mathbf{g}}_k \leftarrow \frac{1}{N_s} \sum_{(\mathbf{x}_n^{(k)}, y_n^{(k)}) \in S_k} f_\theta(\mathbf{x}_n)$
    $\mathbf{v}_k \leftarrow \left\{(N_c - 1)\boldsymbol{\phi}_k - \sum_{l \neq k} \boldsymbol{\phi}_l\right\} - \bar{\mathbf{g}}_k$
  end for
  $\mathbf{M} \leftarrow \text{null}\left(\{\mathbf{v}_k\}_{k \in \{0, ..., N_c - 1\}}\right)$
  for k in $\{0, ..., N_c - 1\}$ **do**
    $Q_k \leftarrow \left\{(\mathbf{x}_n^{(k)}, y_n^{(k)})\right\}$ with $(\mathbf{x}_n^{(k)}, y_n^{(k)}) \in E_i^{(k)}, N_s < n \leq N_s + N_q$
    for $(\mathbf{x}_q, y_q)$ in Q_k **do**
      $\mathbf{g}_q \leftarrow f_\theta(\mathbf{x}_q)$
      $$L_{train} \leftarrow L_{train} + \frac{1}{N_c N_q}\left[d(\boldsymbol{\phi}_k \mathbf{M}, \mathbf{g}_q \mathbf{M}) + \log \sum_{k'} \exp(-d(\boldsymbol{\phi}_{k'} \mathbf{M}, \mathbf{g}_q \mathbf{M}))\right]$$
    end for
  end for
  Update $\theta, \mathbf{\Phi}$ minimizing L_{train} via Adam optimizer
end for