Few-Shot Class-Incremental Learning via Entropy-Regularized Data-Free Replay (Supplementary Material)

Huan Liu[1,2], Li Gu[1], Zhixiang Chi[1], Yang Wang[1,3], Yuanhao Yu[1], Jun Chen[2], and Jin Tang[1]

[1] Noah's Ark Lab, Huawei Technologies
[2] McMaster University, Canada
[3] University of Manitoba, Canada

A Overall Algorithm

Algorithm 1 Entropy-regularized Data-free FSCIL

Require: $\mathcal{T}_i(\cdot;\theta_i)$, $\mathcal{A}_i(\cdot;\theta_{A_i})$, $\mathcal{G}_i(\cdot;\theta_{G_i})$: models
Require: $\alpha, \beta, \lambda_1, \lambda_2$: learning rates
Require: $\{\mathcal{D}^0_{train}, \mathcal{D}^1_{train}, ..., \mathcal{D}^N_{train}\}$: datasets

Base training of $\mathcal{T}_0(\cdot;\theta_0)$ on $\mathcal{D}^0_{train}$ ▷ Session 0.
for $i = 1, ..., N$ **do** ▷ Loop through all the sessions.
  1. Training Generator using entropy regularization.
  while *not converge* **do**
    $z \sim \mathcal{N}(0, I)$ ▷ Sampling noise vector.
    for k iterations **do**
      $\theta_{G_i} \leftarrow \theta_{G_i} - \alpha \frac{\partial \mathcal{L}^*_G}{\partial \theta_{G_i}}$ ▷ Update generator.
    end for
    $\theta_{A_i} \leftarrow \theta_{A_i} - \beta \frac{\partial \mathcal{L}_A}{\partial \theta_{A_i}}$ ▷ Update auxiliary model.
  end while
  2. Learning incrementally with uncertain data.
  initialize $\mathcal{T}_i(\cdot;\theta_i)$ by $\mathcal{T}_{i-1}(\cdot;\theta_{i-1})$
  while *not converge* **do**
    $x^* \leftarrow \mathcal{G}_i(z;\theta_{G_i})$ ▷ Generate data for replaying.
    $y^* \leftarrow argmax(\mathcal{T}_{i-1}(x^*))$ ▷ labeling x^* by old model's activation.
    $\mathcal{D}^i_{train^*} \leftarrow \mathcal{D}^i_{train} \cup \{x^*, y^*\}$ ▷ Update dataset.
    Sample $\{x, y\}$ from $\mathcal{D}^i_{train^*}$
    $\{\theta^b_{i-1}, \theta^l_{i-1}\} = \{\theta^b_{i-1}, \theta^l_{i-1}\} - \lambda_1 \frac{\partial \mathcal{L}_{CE}}{\partial \{\theta^b_{i-1}, \theta^l_{i-1}\}}$ ▷ Fine-tuning old parameters.
    $\theta^{l^*}_i = \theta^{l^*}_i - \lambda_2 \frac{\partial \mathcal{L}_{CE}}{\partial \theta^{l^*}_i}$ ▷ Update new parameters.
  end while
  $\theta_i = \{\theta^b_i, \theta^l_i\}$, where $\theta^l_i = \{\theta^l_{i-1}\, \theta^{l^*}_i\}$
end for
return $\mathcal{T}_N(\cdot;\theta_N)$

In this section, we demonstrate the overall algorithm for a better understanding of our proposed method. At the beginning of each session, we first train a generator given the old model from the previous session. After the training of the generator, we then conduct incremental learning following Section 4.2, where we form a new dataset by combining the novel data with the relabeled replayed data and only use the cross-entropy loss to learn novel classes and avoid forgetting of old classes.