Incremental Few-Shot Learning with Attention Attractor Networks

Seungmin Lee (profile2697@gmail.com; 2013-11420), Dept. of Computer Science and Engineering, Seoul National University

1. Introduction

This paper tried to solve *incremental few-shot learning*. In incremental few-shot learning, we assumes there are base classes and novel classes which are disjoint each other. A model is initially trained using examples from base classes with sufficient labels. Then without re-training on the base classes, we train the model on few-shot labeled novel classes. After trained on novel classes, performance of the model is evaluated on both base and novel classes. If we train the model naively, the performance on base classes is largely dropped (*Catastrophic forgetting*). To avoid this problem, the authors proposed an **Attention Attractor Networks Regularizer** that helps the network not lose too much information about the base classes. They also showed the regularizer can be trained using recurrent back-propagation.

2. Methods

In this method, a model trained using examples from base classes in **Pretrain Stage**. Then the model is trained by repeating **Incremental Few-Shot Learning Stage** and **Meta-Learning Stage**. Additionally. there are four kinds of parameters; a feature extractor, a base classifier parameterized by W_a, a novel classifier W_b, and meta parameters θ_E.

2.1. Pretraining Stage

This step aims to get a good feature extractor and a base classifier parameterized by W_a before training on novel classes. Pretaining stage using typical cross entropy for the base dataset $\{(x_{a,i}, y_{a,i})\}_{i=1}^{N_a} \in \mathcal{D}_a$ where $y_{a,i} \in \{1...K\}$ and $x_{a,i}$ are i-th label and example from the base dataset $\mathcal{D}_a$, respectively.

2.2. Incremental Few-Shot Learning Stage

In this step, the model trained on a episode ξ sampled from the novel dataset $\mathcal{D}_b$ until convergence. The episode ξ is composed of a support set S_b and a query set Q_b, which play the same role as training set and validation set in supervised learning. Additionally, the support set S_b only contains a few labeled samples. In this step, W_a is fixed and W_b is trained using following loss:

$$L^S(W_b, \theta_E) = cross\_entropy(W_b, S_b) + R(W_b, \theta_E). \quad (1)$$

where $R(\cdot, \theta_E)$ is the **attention attractor networks regularizer** parameterized by θ_E. The parameters θ_E are fixed in this stage but trained in the meta-learning stage. The regularizer is used for alleviating catastrophic forgetting at base classes $\mathcal{D}_a$. More details about the regularizer and θ_E are described in **Attention Attractor Networks Regularizer**

Attention Attractor Networks Regularizer The Attention Attractor Networks Regularizer tries to make W_b encode the information about base classes which is contained in W_a. The regularizer is defined as follows:

$$R(W_b, \theta_E) = \sum_{k'=1}^{K'} squared\_mahalanobis\_dist(W_{b,k'}, u_{k'}, \gamma) \quad (2)$$

where K' is number of novel classes, $W_{b,k'}$ is k'-th column of W_b, γ is a learnable parameter which is used for calculating mahalanobis distance, and $u_{k'}$ is *attractor* of a novel class k'.

The attractor $u_{k'}$ is defined as follows:

$$u_{k'} = \sum_k a_{k',k} U_k + U_0 \quad (3)$$

where $U_k = f_\phi(W_{a,k})$ where ϕ is parameters of MLP f. U_k can be interpreted as a learned memory of the base class k. U_0 is a bias term, and $a_{k',k}$ is a normalized pairwise attention between a novel class k' and a base class k. Meanwhile, $a_{k',k}$ has a learnable temparature scala τ. Therefore, θ_E consists of γ, ϕ, U_0, and τ. These parameters are trained in meta-learning stage and are fixed in the incremental few-shot learning stage. Because U_k is the memory of the base class k, $u_{k'}$ contains the information of the base classes. Thus, keeping W_b and $u_{k'}$ close, $R(W_b, \theta_E)$ induces W_b to encode the information that used for base classes.

2.3. Meta-Learning Stage

In this step, the meta parameters θ_E are trained using the episode $\xi = \{S_b, Q_b\}$ and Q_a . θ_E is trained to minimize the expected cross entropy loss for both Q_a and Q_b. Therefore, in this stage, both base and novel classes are used. To train θ_E, Recurrent Back-propagation is used because BPTT needs too many iterations until convergence, and T-BPTT can be unstable depending on T.