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 reduced (*Catastrophic forgetting*). To avoid this problem, the authors proposed an **Attention Attractor Networks Regularizer** that can be interpreted as learned memory for 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** 2.1. Then the model is trained by repeating **Incremental Few-Shot Episodes Stage** 2.2 and **Meta-Learning Stage** 2.3.

2.1. Pretraining Stage

This step aims to get a good feature extractor f and a base classifier parameterized with 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 Episodes

In this step, we will do a few-shot training on the episode $epsilon$ from the few-shot dataset Db. At this time, the episode is composed of support set Sb and query set Qb, which play the same role as training set and validation set in supervised learning. As we progress through each episode, we learn Wb, which is called fast weights, where Wa is fixed, and the loss to Wb consists of:

$Ls(Wb, theta_E) = cross_entropy(Wb, Sb) + R(Wb, theta_E)$

Where R is the attention attractor networks regularizer and parameterized by $theta_E$. $theta_E$ is learned in the meta-learning stage, which is fixed at this stage. As explained earlier, the performance of the model should eventually be good for both base classes and novel classes, which means that the performance for the union Qa + b of the query set Qa for Da and the query set Qb for Db should be high. R is used to alleviate the problem of poor Qa performance due to catastrophic forgetting when only cross entropy is used. In the Incremental Few-Shot Episodes phase, the meta parameters $theta_E$ are fixed. This regularizer term and $theta_E$ are described in the Attention Attractor Networks Regularzier.

2.2.1 Attention Attractor Networks Regularizer

This section describes the Regularizer R. R (Wb, $theta_E$) is defined as

$R(Wb, theta_E) = sum'_k 1, K' squared_m ahalanobis_d istance(Wb, k', u$

In this case, gamma is a learnable parameter and attractor uk is defined as follows.

$uk' = sum_k ak', kUk + U0$

The regularizer term induces the encoding of information used to classify base classes due to attractors when Wb encodes information about a novel class. This is because Uk is the memory encoding base class k, and attractor is the weighted sum of Uk. This term prevents catastrophic forgetting and learns about novel classes.

2.3. Meta-Learning Stage

In this step, you learn the meta parameters $theta_E$ for the episode epsilon. This $theta_E$ is trained to minimize the expected cross entropy loss for Qa + b generated through multiple episodes. At this time, Recurrent Back-propagation can be used to learn $theta_E$. The reason for using this is that BPTT takes too long to learn, and T-BPTT is prone to bias in performance depending on the T value.