Hierarchical Prototypical Networks for Fine-Grained Image Classification

Shubham Kumar Bharti (15807702)
Sunny Kant(15817740)

April 25, 2019

1 Introduction

Prototypical Network have been shown to achieve better classification results on the limited data regimes of Few-shot learning and Zero-shot learning [3]. There effectiveness have been evaluated on more general Image-Net dataset. In this project, we aim to develop Prototypical Networks to improve Fine-grained and Few-shot fine-grained image classification task. Fine-grained image classification suffers from large intra class variations and subtle inter class differences which makes it hard to learn a good discriminator. We aim to generate fine-grained prototypes by introducing hierarchical cross-entropy loss and other means to improve inter-class prototype variances at different hierarchical levels. We tried both single category classification and joint multi category classification to further improve generalization performance of the network. We have used four prominent Fine-grained classification dataset namely - CUBS Birds dataset, Stanford Cars dataset, Oxford FGVC Aircrafts dataset and Stanford Dogs dataset dataset.

2 Related Works

In this section, we describe some recent previous works on Prototypical Networks and Fine-Grained image classification that have been shown to work

well. Prototypical Networks has been used recently for Few-shot classification [3]. In case of limited training data, classification through learned prototypes corresponding to each class has shown better results. In an another work, Two feature extractors have been used in Bilinear CNN models [2] to capture local pairwise feature interactions. Outputs obtained from two CNNs are multiplied using outer product and then pooled to obtain bilinear feature vector for classification. Recently, Hierarchical Semantic Embedding (HSE) [1] has been used to predict the score vector in each level in the hierarchy from coarse to fine level. HSE framework has shown its superiority in Caltech-UCSD birds dataset with the four-level category hierarchy.

Terminologies

- We would refer 5 coarse classes by the term 'class' itself and finer classes within a coarse class by the term 'subclass' of a 'class'.
- When we talk of a prototype, it refers to the prototype of a subclass(the fine-grained class). The prototype of a class(coarse class) is defined by the mean of the prototypes of all subclasses in that class.

2.1 Prototypical Networks

Data points associated to each class would form a cluster in new non-linear embedding space. The mean of the new embedding associated to each cluster can represent the prototype of that class. Classification can be performed based on various distance measures. Mapping of input space to non-linear embedding space is done by a prototypical neural network. Training data of each class is divided in support points and query points to train the model to learn to map query data point closer to the prototype calculated by mean of embedded support vectors of the each sub-classes. Embedded vector of support points is used to calculate prototype of a sub-class and query points are used to learn the neural network. Considering embedding function $f_\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M$, prototype of k^{th} class is mean of its embedded support points.

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi \left(\mathbf{x}_i \right)$$

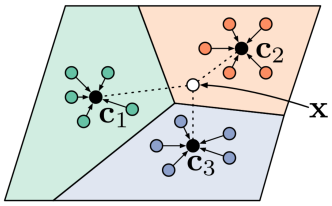

Figure 1: Prototypical Network

2.2 Fine-Grained Image Classification Task

Fine-Grained Classification is the classification of classes that are difficult to distinguish like species of birds, models of cars or aircrafts etc. It is classification of different subclasses of particular class or classes. As subclasses share common features, so learning model that captures distinctive features in such cases is generally difficult than coarse classification.

3 Our Approach

We modified prototypical networks for adapt to fine-grained image classification. We have employed various strategies and loss functions to take care that good latent feature space is learned that can help to classify an image by associating with the label of its closest prototype. The details are given below.

3.1 Training Algorithm

Initialize the Prototypical Neural Network
↪ prototypical network maps an image to the latent space

Sample N_C number of subclasses from the dataset
▶ Finding Prototypes using Support set
for *each subclass in the* $[N_C]$ **do**
  Sample support set and query set of that dataset;
  Obtain the subclass prototypes using support set;
end
▶ Optimizing Network using Query set
for *each subclass in the* $[N_C]$ **do**
  for *each sample in the query set* **do**
    Assign the sample to its closest prototype in the latent space.
    Calculate the cross entropy loss
  end
end
Update the Network to minimize the sum of Cross Entropy loss.

3.2 Optimizing Prototype Variances

We tried two different settings to optimize the objective to learn a better feature representation space.

3.2.1 Maximizing all the pairwise Prototypical distance of all subclasses

This setting ensures that variance within subclass samples are small and subclass prototypes themselves are spread far apart. ↪ After the gradient update from previous step

- Recalculate the prototypes using Support Set
- Calculate S = sum of pairwise distances of all prototypes
- Update the Network to maximize S

Note : This combined with previous optimization step forms an alternating optimization schedule.

3.2.2 Maximizing the pairwise Prototypical distance of class prototypes

In this method, we try to keep the class prototypes far apart and subclass prototypes within each class closer to each other.

↪ After the gradient update from previous step

- Recalculate the prototypes using Support Set
- Calculate S = sum of pairwise distances of all class prototypes
- Calculate T_i = sum of pairwise distances of all prototypes within individual classes(T_i denotes the sum for i^{th} class).
- Update the Network to maximize $\alpha S + \beta \sum_{i=1} IT_i$

3.3 Hierarchical Cross-Entropy Loss Function

▶ Our Hierarchical Cross-Entropy loss is defined as

$$loss = -\Big[\alpha \sum_{c=1}^{M} y_{o,c} \log p_{o,c} + (1-\alpha) \sum_{d \notin S_o} \log\left(1 - p_{o,d}\right)\Big]$$

where S_o is the set of all the sister subclasses of correct subclass of object o.
and α is a weighting parameter

- If a image gets classified to a subclass label in other class higher loss is incurred.
- If a image gets classified to a subclass label of same parent class less loss is incured.

4 Dataset

We will use dataset of following 5 coarse classes and corresponding fine classes. We will sample our train and test dataset from standard train and test dataset respectively.

1. CUBS Birds Dataset
   This dataset with consists of photos of 200 different bird species. [4]

2. FGVC-Aircraft dataset
   The dataset contains 10,200 images of aircraft, with 100 images for each of 102 different aircraft models.

3. Stanford Cars Dataset
   The dataset contains 16,185 images of 196 classes of cars.

4. Stanford Dogs Dataset
   The dataset consists of 120 dog classes with approximately 150 images per class.

5 Results

We trained various encoding architecture including various fine-tuned Image Net classification network and manually curated networks similar to the ones mentioned in original paper. We report the best accuracy which we have obtained using our network on different dataset. Due to time constraints we were not able to perform an exhaustive set of experiments on our dataset. We mention our early results(not completely optimized) below.

Training Classes Individually

1. Aircraft : 0.63 (using conv_blocks with resnet34)
2. Birds : 0.52 (using conv_blocks with resnet34)
3. Dogs : 0.35 (less trained conv_block only)
4. Cars : 0.31 (less trained conv_blocks only)

Training Classes Jointly(using conv_blocks with resnet34)

1. Aircrafts : 0.25
2. Birds : 0.30
3. Dogs : 0.27
4. Cars : 0.21

6 Challenges & Further Improvements

1. Due to limited GPU resources, we were forced to consider very small(10 support & 5 query) samples from each sub-classes in the data. A better stochastic mean could be obtained by considering higher number of samples from the corresponding data distribution.

2. Figuring out a good encoding prototypical(that capture our inductive bias to transform such a huge variety of images into a good latent space) also appeared to be time-consuming - training on each network takes around 20 hours. Training on all (coarse) classes was even painful as the data size became huge.

3. Further improvements can be made in prototypical loss function to enforce a particular subclass to embed latent features closer to sub classes within same class than to the sub-classes of other class.

References

[1] Tianshui Chen, Wenxi Wu, Yuefang Gao, Le Dong, Xiaonan Luo, and Liang Lin. Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding. *CoRR*, abs/1808.04505, 2018.

[2] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1449–1457, Washington, DC, USA, 2015. IEEE Computer Society.

[3] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, 2017.

[4] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.