이 학 석 사 학 위 논 문

Adaptive Layerwise Prototype Networks

적응형 층별 원형 네트워크

2020년 2월

서울대학교 대학원

통계학과

주 여 진

Adaptive Layerwise Prototype Networks

적응형 층별 원형 네트워크

지도교수 김 용 대

이 논문을 이학석사 학위논문으로 제출함

2019년 12월

서울대학교 대학원

통계학과

주 여 진

주여진의 이학석사 학위논문을 인준함

2019년 12월

위 원 장 이 재 용 (인)

부위원장 김 용 대 (인)

위 원 오 희 석 (인)

Abstract

Adaptive Layerwise Prototype Networks

Yeojin Joo
The Department of Statistics
The Graduate School
Seoul National University

With the development of deep learning, many researches in the field of computer vision, such as object recognition, are showing good results. However, deep learning requires a lot of data and time compared to humans. Humans can solve simple classification problems with just one example, but the machine needs many examples to optimize the parameters. Thus, few-shot learning emerged from the discussion of creating a model that could adapt quickly to new challenges with less data. Prototypical networks (Snell et al. (2017)) are well known as a representative metric-based model of few-shot learning. It sends each input to the embedding space, where the same classes are closer together and the other classes are farther apart. In this paper, we develop this networks and make networks that perform better. We build a model that uses layered prototypes to use not only high-level features but also mid-level and low-level features. In addition, each prototype is given a different weight for the final model, and the weights are trained together so that the network can adapt to the problem.

Keywords : *few-shot learning, meta-learning, prototypical networks, multi-level features*

Student Number : 2018-25706

Contents

Chapter 1

Introduction

Deep learning has played a large role in solving real-world problems such as autonomous driving and machine translation. In particular, deep learning is doing much of what humans do and expected to replace more people in the near future. Furthermore, people hope that deep learning model will act like the way the human brain works.

However, there is a big difference between deep learning and the human brain. It is the amount of data and time required for learning. For example, suppose that a human and a machine are given a problem of classifying cats and birds images. In this case, a human can solve the problem with only one training image for cats and birds. The human brain finds differences from the images of cats and birds, and answers the question of distinguishing cats from birds based on these differences. Like this, the human brain is well designed to solve problems and find answers. However, what about the machine? It will be hard to be sure that the machine will find the right answer in the same

situation. Let's explain it with the number of parameters, one of the factors that determine the performance of a machine. The fewer the parameters, the less information that can be taken from the image, and the more parameters, the more training images are needed to optimize the parameters. Therefore, machines require more training data than humans.

Recently, a lot of research has been done to create a model that can adapt to new problems faster, with less information, like the real human brain. This field of research is called meta learning or few-shot learning.

In chapter 2, we review the basic concepts of few-shot learning and prototypical networks (Snell et al. (2017)), one of the few-shot learning methods. And chapter 3 introduces our proposed model, Adaptive Layerwise Prototype networks, we explain the motivation and the algorithm of the model. In chapter 4, we describe the process and result of model experiment with data. After mentioning related work in chapter 5, the conclusion of the study is discussed in chapter 6.

Chapter 2

Review of Few-shot Learning, Prototypical Networks

2.1 Few-shot Learning

Few-shot learning is a methodology that allows a machine to learn the way the human brain learns, and is often used interchangeably with the concept of meta learning. The human brain, unlike a machine, has the ability to solve new problems with only few examples, even with just one, and few-shot learning studies have begun in the hope that AI will behave more and more like the real human brain.

The goal of few-shot learning is to make learning faster for new tasks. Therefore, few-shot learning takes a slightly different approach from traditional machine learning. When evaluating the performance of machine learning, it is enough to compare accuracy or loss with a single test set. On the

other hand, when evaluating the performance of meta-learning, we need to make multiple train sets and test sets to check that the model can adapt quickly to new tasks. These multiple train sets and test sets are called meta-train and meta-test, respectively. Each set is called an episode, which is divided into a support set and a prediction set. These episodes are sequentially trained and tested to check that the model is performing well for the new task.

In order to compare the performance of various techniques in few-shot learning, it is necessary to specify the number of data used for learning and the number of label to classify. Therefore, the model performance of few-shot learning must be specified together as k-shot n-way, where k-shot n-way task means that the support set contains k labeled data for each N classes.

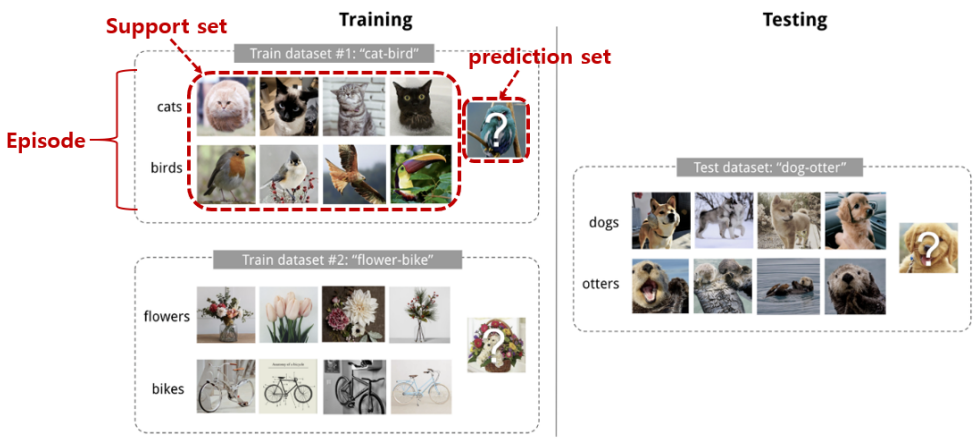

Figure 2.1: The structure of few-shot learning

2.2 Prototypical Networks

Few-shot learning has evolved in three major ways: matching networks (Vinyals et al. (2016)), Model-agnostic Meta-learning (MAML) (Finn et al. (2017)), and prototypical networks (Snell et al. (2017)). Among these, prototypical networks showed the best performance, so in this study, we chose prototypical networks as the base model and transformed the networks.

Table 2.1: Few-shot classification accuracies

Methods	**20-way Omniglot**		**5-way miniImageNet**	
	1-shot	5-shot	1-shot	5-shot
Matching Nets	88.2%	97.0%	43.56 ± 0.84%	55.31 ± 0.73%
MAML	95.8%	98.9%	48.70 ± 1.84%	63.15 ± 0.91%
Prototypical Nets	96.0%	98.9%	49.42 ± 0.78%	68.20 ± 0.66%

Prototypical networks (Snell et al. (2017)) is one of the representative models of the metric-based meta learning model. The basic concept is similar to the nearest neighbors algorithm and kernel density estimation. First, they embed each input value x_i into a low dimensional vector $f_\theta(x_i)$. Then, use the embedding vectors $f_\theta(x_i)$ to define the prototype feature vector v_c for each label. Next, the class distribution for a given test input x is calculated using the distance between the embedded vector $f_\theta(x_i)$ and the prototype feature vector v_c. That is, $P(y = c|x)$ is equivalent to taking a softmax over the inverse of the distance between the test data embedding vector and the prototype feature vector. Here, the distance can be any distance function that can be differentiated, and they usually use squared Euclidean distance because it gives the best accuracy, and use negative log-likelihood as a loss

function.

$$
\begin{aligned}
x_i &\rightarrow f_\theta(x_i) \\
v_c &= \frac{1}{|S_c|} \sum_{(x_i, y_i) \in S_c} f_\theta(x_i) \\
P(y = c|x) &= softmax(-d_\phi(f_\theta(x), v_c)) = \frac{exp(-d_\phi(f_\theta(x), v_c))}{\sum_{c' \in C} exp(-d_\phi(f_\theta(x), v_{c'})}
\end{aligned}
\tag{2.1}
$$

Chapter 3

Adaptive Layerwise Prototypical Networks

3.1 Motivation

In deep learning models such as cnn, it is known that the level of features that can be extracted from each layer varies depending on the depth of the layer (Krizhevsky et al. (2012)).

Figure 3.1 shows the feature of each layer extracted through the filter in the cnn model. Low-level features such as edges and faces are extracted from the shallow layer, and mid-level features such as wheels and large parts are extracted from the middle layer. Visualization also shows that high-level features, such as headlight lamps and small parts, are extracted from deeper layers. Deep models usually extend the last layer and are fully connected to

produce the final value. That is, use only high-level features. Sometimes both mie-level and high-level features are used.

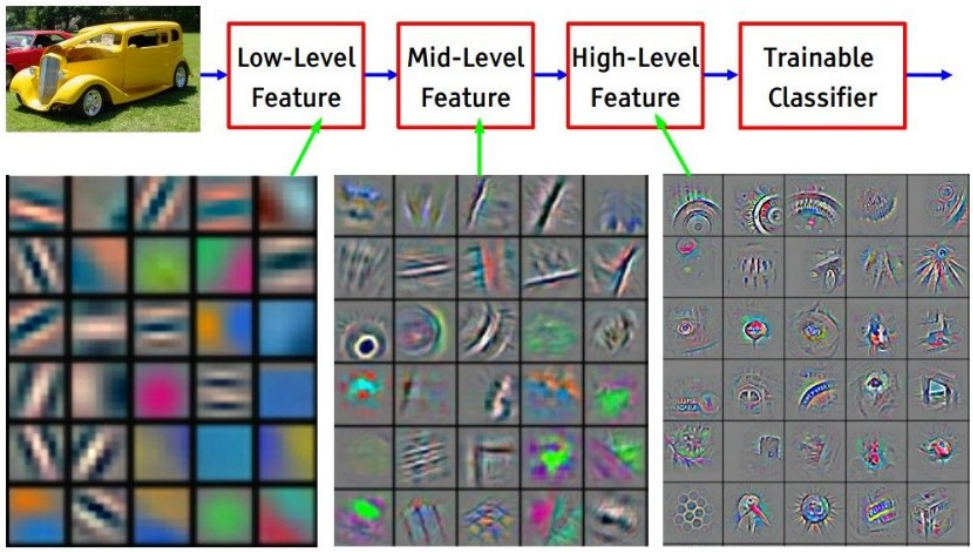

Figure 3.1: Level of feature in each layer

Because the goal of few-shot learning is to train faster with less data, networks must be able to extract a lot of information from the limited data and use it all at once.

Therefore, in order to develop existing prototypical networks, we propose a model that can use not only high-level features but also low-level and mid-level features. In addition, we propose networks that show better performance by deepening layers and increasing dimension size than the cnn structure used in existing networks.

3.2 Model

We construct the CNN structure as Figure 3.2 and obtain the embedding vector from each layer and calculate the probability that each class belongs to the extracted embedding vector. Then there are three probability values, these are weighted together to calculate the final probability. At this time, the weight is also treated as a parameter and learned together.

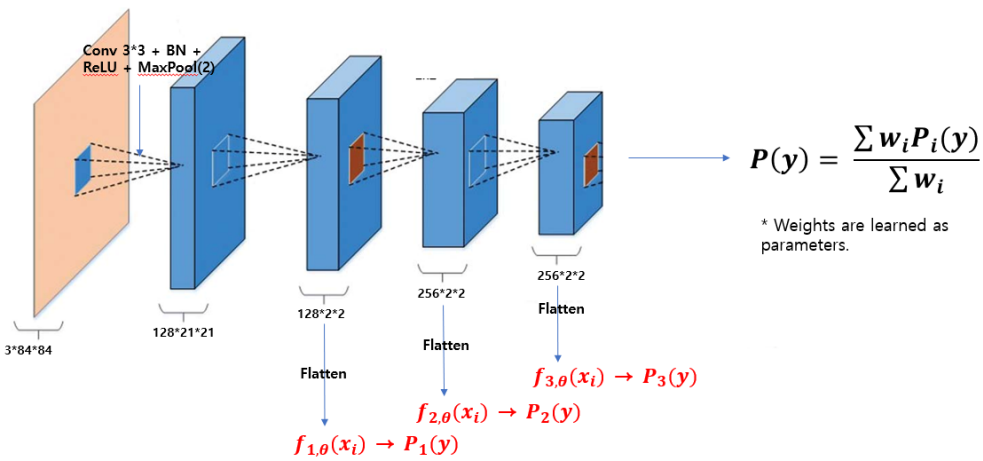

Figure 3.2: Adaptive Layerwise Prototype Nets' structure for miniImagenet

Algorithm 1 Adaptive Layerwise Prototype Networks (N-shot K-way), N is the number of examples, K is the number of classes, N_S is the number of support examples, N_Q is the number of query examples per episode. K' is the number of classes in training set ($K \leq K'$), $Samp(S, N)$ denotes N elements chosen randomly without replacement from set S.

Input: Train set $D = \{(x_1, y_1), ..., (x_N, y_N)\}$, $D_k = \{(x_i, y_i)|y_i = k\}$

Output: The loss L calculated from training episodes.

$V \leftarrow Samp(\{1, ..., K'\}, K)$

for i in $\{1, 2, ..., K\}$ **do**

$S_i \leftarrow Samp(D_{V_i}, N_S)$

$Q_i \leftarrow Samp(D_{V_i} \setminus S_i, N_Q)$

$f_\theta(x_i) = \frac{\sum w_j * f_{\theta_j}(x_i)}{\sum w_j}$

$v_i = \frac{1}{K} \sum_{(x_j, y_j) \in S_i} f_\theta(x_j)$

end for

$L \leftarrow 0$

for i in $\{1, 2, ..., K\}$ **do**

for (x, y) in Q_i **do**

$L \leftarrow L + \frac{1}{K * N_Q}(d(f_\theta(x), v_i) + log \sum_{\neq i} exp(-d(f_\theta(x), v_i))$

end for

end for

Chapter 4

Experiments

4.1 Omniglot Few-shot Classification

Omniglot is one of the most frequently used data for few-shot learning study, and contains written texts from worldwide languages. A total of 1623 characters in 50 languages were written by 20 different people. Therefore, the database contains 20*1623 data.

The input size is 1 * 64 * 64 and 3 * 3 filter is applied to each layer. After batch normalization for regularization, ReLU was used as an activation function and 2 * 2 maxpooling was applied. Likewise, each layer was calculated in the same way, and the total 3 layers were used for the network. At this time, the dimension of each embedding vector is 512, 1024, 2048. The results show that our model outperforms prototypical nets in both 1-shot and 5-shot.

Table 4.1: performance on Omniglot

	1-shot 20-way	5-shot 20-way
Prototypical Nets	95.12	98.09
Adaptive Layerwise Proto Nets	**96.29**	**98.87**

4.2 miniImagenet Few-shot Classification

miniImagenet is also frequently used for few-shot learning (Vinyals et al. (2016)). It was sampled from Imagenet data. A total of 100 classes were randomly sampled and 600 examples corresponding to each class were randomly sampled.

The input size is 3 * 84 * 84 and 3 * 3 filter is used to each layer as in Omniglot. and After batch normalization was performed for regularization, and ReLU was used as an activation function and then, we took 2 * 2 max-pooling, and by repeating this process, we obtained the total 3 layers for the network. The dimension of each embedding vector is 512, 1024, 2048. The results show that our model's performance was improved in 1-shot but not in 5-shot.

Table 4.2: performance on miniImagenet

	1-shot 5-way	5-shot 5-way
Prototypical Nets	48.32	**66.93**
Adaptive Layerwise Proto Nets	**50.14**	66.30

Chapter 5

Related Work

Many few-shot learning studies, including prototype networks, are based on metric-based learning. The goal of metric-based leanrning is to learn embedding spaces that allow data from the same label to be close together and data from different labels to be far from each other. Li et al. (2019) proposed DN4, which consists of a cnn-based embedding module that learns local descriptors and an image-to-class module that learns similarities between query images and classes. Next, Davis et al. (2019) proposed a model to perform localization and classification based on prototypical nets, but there is a limit to requiring data with bounding boxes. Lifchitz et al. (2019) omit global average pooling at the end of the model to perform classification in dense spaces. Finally, Kim et al. (2019) proposed a method that can be applied to the real world, especially the logo and road sign classification problem.

Chapter 6

Conclusion

In the experimental results, the performance of both 1-shot and 5-shot was improved compared to prototypical nets in omniglot data, but only 1-shot was improved in miniImagenet data. Also, the fewer shots, the greater the degree of performance improvement. Our model has the advantage of using various levels of features from low-level to high-level compared to the existing model. This, in turn, means how important the additional mid-level and low-level features used in the classification model will determine the performance differences between the prototypical nets and our model.

The omniglot data has a simpler image structure than miniImagenet data. Therefore, low-level features helped improve performance in simpler omniglots, but did not seem to help much in miniImagenet. Also, when using less data, it seems that performance is improved because using various levels of features can utilize more information.

Reference

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4080-4090.

Vinyals, Oriol, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems.*

Wertheimer, Davis, and Bharath Hariharan. 2019. Few-Shot Learning with Localization in Realistic Settings. *Computer Vision and Pattern Recognition.*

Finn, Chelsea, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org.*

Junsik Kim, Tae-Hyun Oh, Seokju Lee, Fei Pan, In So Kweon. 2019. Variational Prototyping-Encoder: One-Shot Learning with Prototypical Images. *Computer Vision and Pattern Recognition.*

Krizhevsky, Alex, Ilya Sutskever, Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in neural information processing systems.*

Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, Andrei Bursuc. 2019. Dense Classification and Implanting for Few-Shot Learning.*Computer Vision and Pattern Recognition.*

Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. 2019. Revisiting local descriptor based image-to-class measure for few-shot learning. *Computer Vision and Pattern Recognition*, pages 7260–7268.

국 문 초 록

적응형 층별 원형 네트워크

딥러닝의 발전 덕분에, 사물 인식 등 컴퓨터 비전 분야의 많은 연구들이 최근 좋은 성과를 보여주고 있다. 그러나, 딥러닝은 간단한 문제를 해결하는 데에도 인간에 비해 꽤 많은 데이터와 시간을 요구한다. 인간은 단 한 개의 예제만으로도 간단한 분류문제를 해결할 수 있지만, 기계는 파라미터들을 최적화시키기 위해 많은 예제를 필요로 한다. 따라서, 적은 데이터로 빠르게 새로운 과제에 적응할 수 있는 모델을 만들어보자는 논의에서 few-shot learning이 출현하였다. few-shot learning의 대표적인 metric-based model로는 prototypical networks가 널리 알려져있다. prototypical networks는 각 input을 embedding space로 보내고, 이 공간상에서 동일한 클래스끼리는 가깝게, 다른 클래스끼리는 멀게끔 한다. 이 논문에서는, prototypical networks를 발전시켜 더 좋은 성능을 보여주는 네트워크를 만든다. 이 networks를 발전시키기 위해, 우리는 특히, high-level의 feature만 사용하는 기존의 모델에서 더 나아가, mid-level과 low-level의 feature까지 사용할 수 있도록 층마다의 prototype을 활용한 모델을 구축한다. 또한, 각 prototype마다 최종 모델에 활용되는 weight를 다르게 주고, 이 weight를 함께 학습시킴으로써 네트워크가 문제에 잘 적응할 수 있도록 한다.

주요어 : 퓨샷러닝, 메타러닝, 원형 네트워크, 다차원 수준 속성

학 번 : 2018-25706