IMPROVED PROTOTYPICAL NETWORK WITH DATA AUGMENTATION

Technical Report

Dongchao Yang[1], *Yuexian Zou*[1,*], *Fan Cui*[2], *Yujun Wang*[2]

[1] Peking University, School of ECE, Shenzhen, China
[2] Xiaomi Corporation, Beijing, China
dongchao98@stu.pku.edu.cn, zouyx@pku.edu.cn, cuifan@xiaomi.com, wangyujun@xiaomi.com

ABSTRACT

In this technical report, we describe our few-shot bioacoustic event detection methods submitted to Detection and Classification of Acoustic Scenes and Events Challenge 2022 Task 5. We follow our previous work, and further improve our model through data augmentation strategy. Specifically, we analyze the reason why Prototypical networks cannot perform well, and propose to use transductive inference for few shot learning. Our method maximizes the mutual information between the query features and their label predictions for a given few-shot task, in conjunction with a supervision loss based on the support set. Furthermore, we use multiple data augmentation strategies to improve the feature extractor, including time and frequency masking, mixup, and so on. Experimental results indicate our model gets better performance than baseline, and F1 score is about 51.9% on evaluation set.

Index Terms— few shot learning, transductive inference, sound event detection, data augmentation

1. INTRODUCTION

Few-shot learning [1, 2, 3] is a highly promising paradigm for sound event detection. It is also an extremely good fit to the needs of users in bioacoustics, in which increasingly large acoustic datasets commonly need to be labelled for events of an identified category (e.g. species or call-type), even though this category might not be known in other datasets or have any yet-known label. Sound event detection (SED) aims to classify and localize all pre-defined sound events (*e.g.*, train horn, car alarm) within an audio clip, which has been widely studied [4, 5, 6, 7, 8]. DCASE2022 task5 aims to detect the target sound but only provides the five annotated data.

Few-shot learning describes tasks in which an algorithm must make predictions given only a few instances of each class, contrary to standard supervised learning paradigm. The main objective is to find reliable algorithms that are capable of dealing with data sparsity, class imbalance and noisy or busy environments. Few-shot learning is usually studied using N-way-K-shot classification, where N denotes the number of classes and K denotes the number of examples for each class.

Few-shot learning tasks have been increasingly studied in literature and often rely on meta-learning approaches including MAML (Model-Agnostic Meta-Learning) [9], Prototypical network [10], Relation network, and so on. Most such works are done in computer vision [11, 3] or natural language recognition [12] while very little work has been done in audio-related tasks. In [13], authors compare different few shot methods on acoustic event detection, which shows Prototypical network gets better performance. In the challenge of DCASE2021 task5, the official baseline [14] also chooses Prototypical network.

In this paper, we follow our previous works [15, 16] to dynamic update the class prototypes using network. Furthermore, we explore using data augmentation strategies to improve the ability of feature extractor.

1.1. Prototypical networks

Most of the existing approaches within the FSL framework are based on the "learning to learn" paradigm or meta-learning, where the training set is viewed as a series of balanced tasks (or episodes), so as to simulate test-time scenario. Prototypical network [10] is one of successful examples using meta-learning. In this task, the baseline chooses Prototypical network. The main idea of Prototypical network is using few-shot prototypes as class center, and then we judge unlabeled data belongs to which class according to their L_2 distance to class center. Few-shot prototypes are computed as the mean of embedded support examples for each class.

According to previous description, we can find that a good prototype representation is very important. After training, we can believe our network can extract class-wise information from data, but the problem is most of the samples are incomplete, such as incomplete samples, background interference or fuzzy details, so that some representative attribute characteristics are lost. This is why Prototypical networks cannot perform well on some tasks. In this challenge, evaluation set give 8 different audio files, we find that if the labeled part (support sets) has good quality, the results of Prototypical network is very good. So the key point is to find a good prototype representation for these incomplete support samples.

1.2. Motivation

As previous discussion, incomplete support sets data will lead to prototype cannot represent the category center. One way to solve this problem is finding supplementary information, which can help prototype representation as close to the true category center as possible. In this report, we follow our previous works to update the prototype representations.

Furthermore, we also note that Feature extractor is very import, thus we use multiple data augmentation strategies to improve the ability. In this work, time and frequency masking [17], mixup, and label smoothing are used.

2. METHODS

Build classifier based on class prototypes For a given few-shot task, with a support set S and a query set Q, let X denote the random variables associated with the acoustic features within $S \cup Q$ and let $Y = \{1, 2, ..., K\}$ be the random variables associated with the labels. Let $f_\phi : X \rightarrow Z \subset R^d$ denote the encoder (*i.e.*, feature extractor) function of a deep neural network, where ϕ denotes the trainable parameters, and Z stands for the set of embedded features. The encoder is firstly trained from the base training set X_{base} using the standard cross-entropy loss. Next, for each specific few-shot task, we define a classifier, parametrized by a weight matrix $\boldsymbol{W} = [\boldsymbol{w}_1, ..., \boldsymbol{w}_K] \in R^{K \times d}$. The posterior distribution over labels given features is defined by $p_{ik} = P(Y = k|X = \boldsymbol{x}_i; \boldsymbol{W}, \phi)$. The marginal distribution over query labels is defined by $\hat{p}_k = P(Y_Q = k; \boldsymbol{W}, \phi)$. p_{ik} and $\hat{p}_k$ are calculated as formula (1).

$$p_{ik} = \frac{exp(\boldsymbol{w}_k \cdot \boldsymbol{z}_i)}{\sum_{c=1}^{K} exp(\boldsymbol{w}_c \cdot \boldsymbol{z}_i)}, \hat{p}_k = \frac{1}{Q} \sum_{i \in Q} p_{ik} \tag{1}$$

where $\boldsymbol{z}_i = \frac{f_\phi(\boldsymbol{x}_i)}{||f_\phi(\boldsymbol{x}_i)||_2}$ denotes L2-normalized embedded features. For each task, weights $\boldsymbol{W}$ are initialized by the class prototypes of the support set, as follows

$$\boldsymbol{w}_k = \frac{1}{|S_k|} \sum_{\boldsymbol{x}_i \in S_k} f_\phi(\boldsymbol{x}_i) \tag{2}$$

In this paper, we only need to judge whether the audio frame is a positive sample, so we set $K = 2$.

Updating classifier To update the weight matrix $\boldsymbol{W}$, for each single few-shot task, we propose a loss function with two complementary terms: (1) a standard cross-entropy loss on the support set; (2) a mutual-information loss, which includes a conditional entropy loss and a marginal entropy loss.

$$L_w = \lambda_{CE} \cdot CE - \hat{I}(X_Q; Y_Q) \tag{3}$$

$$CE = -\frac{1}{|S|} \sum_{i \in S} \sum_{i=1}^{K} y_{ik} log(p_{ik}) \tag{4}$$

$$\hat{I}(X_Q; Y_Q) = -\sum_{k=1}^{K} \hat{p}_k log \hat{p}_k + \frac{1}{|Q|} \sum_{i \in Q} \sum_{k=1}^{K} p_{ik} log(p_{ik}) \tag{5}$$

where CE denotes the cross entropy loss function, y_{ik} denotes the true label of the sample in the support set, p_{ik} denotes the prediction result. In our experiments, λ_{CE} is set as 0.1. $\hat{I}(X_Q; Y_Q)$ denotes the mutual information between the query samples and their latent labels. It is a combination of two terms, the first term is the empirical label-marginal entropy, denoted as $\hat{H}(Y_Q)$, while the second term is an empirical estimate of the conditional entropy of labels given the query acoustic features, denoted as $\hat{H}(Y_Q|X_Q)$. $\hat{H}(Y_Q|X_Q)$ aims at minimizing the uncertainty of the posteriors at unlabelled query samples, thereby encouraging the model to output confident predictions. This entropy loss is widely used in the context of semi-supervised learning (SSL) [18, 19], as it models effectively the cluster assumption: the classifier's boundaries should not occur at dense regions of the unlabelled features. The label-marginal entropy regularizer $\hat{H}(Y_Q)$ encourages the marginal distribution of labels to be uniform.

Note that we only update the weight matrix $\boldsymbol{W}$ in this step, while the feature extractor is fixed. Our experimental results also show that simultaneously updating feature extractor f_ϕ and weight matrix $\boldsymbol{W}$ does not offer better performance.

Table 1: F-score comparison of different methods on DCASE 2022 task5 evaluation set.

Method	precision	recall	F-score
Baseline	36.34	24.96	29.59
ours	**59.2**	**46.34**	**51.99**

3. EXPERIMENT

3.1. Dataset and metrics

Dataset The dataset is from DCASE2022 task5 evaluation set. **Metrics** For all the experiments, we use the event-based F-measure as the evaluation metric, which is one of the most commonly used metrics for sound event detection.

3.2. Setups

For training, we follow the same settings as the baseline [14]. Our feature extractor as same as baseline backbone, which includes 4 convolution layers. Specifically, the input i down-sampled to 22.05kHz and applied a Short Time Fourier Transform (STFT) with a window size of 1024, followed by a Mel-scaled filter bank on perceptually weighted spectrograms. This results in 128 Mel frequency bins and around 86 frames per second. The input frames are normalized to zero-mean and unit variance according to the training set. The Adam optimizer [20] is used for a total of 50 epochs, with an initial learning rate of 0.0001. The learning rate decays linearly from epoch 10. The difference is that we never use meta-learning training strategy, otherwise we directly train feature extractor by cross entropy loss, and the data augmentation is used in training process. Furthermore, we do not use any ensemble approaches.

For inference, to update the parameter of classifier, the Adam optimizer [20] is used for 10 epochs, with an initial learning rate of 1×10^{-5}. The last epoch predicted results are used as our final predicted results.

3.3. Experimental results

Table 1 shows the experimental results, which indicate our proposed method is very useful.

4. CONCLUSIONS

In this report, we follow our previous methods, and further use data augmentation strategies to improve the ability of feature extractor. Experimental results indicate the effectiveness of our methods.

5. REFERENCES

[1] E. G. Miller, N. E. Matsakis, and P. A. Viola, "Learning from one example through shared densities on transforms," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR*, vol. 1. IEEE, 2000, pp. 464–471.

[2] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of*

The Annual Meeting of The Cognitive Science Society, vol. 33, no. 33, 2011.

[3] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, vol. 2. Lille, 2015.

[4] H. Dinkel, M. Wu, and K. Yu, "Towards duration robust weakly supervised sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 887–900, 2021.

[5] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound event detection of weakly labelled data with cnn-transformer and automatic threshold optimization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2450–2460, 2020.

[6] D. Yang, H. Wang, Y. Zou, and C. Weng, "Detect what you want: Target sound detection," *arXiv preprint arXiv:2112.10153*, 2021.

[7] D. Yang, H. Wang, Y. Zou, and W. Wang, "A two-student learning framework for mixed supervised target sound detection," *arXiv preprint arXiv:2204.02088*, 2022.

[8] D. Yang, H. Wang, Z. Ye, Y. Zou, and W. Wang, "Radur: A reference-aware and duration-robust network for target sound detection," *arXiv preprint arXiv:2204.02143*, 2022.

[9] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.

[10] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 4077–4087.

[11] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 2016.

[12] M. Yu, X. Guo, J. Yi, S. Chang, S. Potdar, Y. Cheng, G. Tesauro, H. Wang, and B. Zhou, "Diverse few-shot text classification with multiple metrics," *arXiv preprint arXiv:1805.07513*, 2018.

[13] B. Shi, M. Sun, K. C. Puvvada, C.-C. Kao, S. Matsoukas, and C. Wang, "Few-shot acoustic event detection via meta learning," in *ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 76–80.

[14] https://github.com/c4dm/dcase-few-shot-bioacoustic/tree/main/baselines.

[15] D. Yang, H. Wang, Y. Zou, Z. Ye, and W. Wang, "A mutual learning framework for few-shot sound event detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 811–815.

[16] D. Yang, H. Wang, Z. Ye, and Y. Zou, "Few-shot bioacoustic event detection: A good transductive inference is all you need," DCASE2021 Challenge, Tech. Rep, Tech. Rep., 2021.

[17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[18] Y. Grandvalet, Y. Bengio, *et al.*, "Semi-supervised learning by entropy minimization." in *CAP*, 2005, pp. 281–296.

[19] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *arXiv preprint arXiv:1905.02249*, 2019.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.