STAGED TRAINING STRATEGY AND MULTI-ACTIVATION FOR AUDIO TAGGING WITH NOISY AND SPARSE MULTI-LABEL DATA

Kexin He**, Yuhan Shen*†*, Wei-Qiang Zhang***, Jia Liu**

*Beijing National Research Center for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
†Khoury College of Computer Sciences, Northeastern University, Boston 02115, MA, USA
hekexinchn@163.com, shen.yuh@husky.neu.edu, {wqzhang,liuj}@tsinghua.edu.cn

ABSTRACT

Audio tagging aims to predict whether certain acoustic events occur in the audio clips. Due to the difficulty and huge cost of obtaining manually labeled data with high confidence, researchers begin to focus on audio tagging using a small set of manually-labeled data, and a larger set of noisy-labeled data. Besides, audio tagging is a sparse multi-label classification task, where only a small number of acoustic events may occur in an audio clip. In this paper, we propose a staged training strategy to deal with the noisy label, and adopt a sigmoid-sparsemax multi-activation structure to deal with the sparse multi-label classification. This paper is an improvement and extension of our previous work for participation in Task 2 of Detection and Classification of Acoustic Scenes and Events (DCASE) 2019 Challenge. We evaluate our methods on the identical task, and achieve state-of-the-art performance, with a lwlrap score of 0.7591 on official evaluation dataset.

Index Terms— Audio tagging, noisy label, staged training strategy, multi-activation structure, DCASE2019 Challenge

1. INTRODUCTION

Audio tagging aims to identify the presence or absence of sound events in the audio clip without predicting the onset and offset times of these events. Recently, audio tagging have attracted attentions of both academia and industries due to its wide application prospects, such as query-based sound retrieval [1], smart homes [2] and acoustic surveillance [3].

In the early works, researchers utilize the Gaussian Mixture Model (GMM) and Support Vector Machine (SVM) for audio tagging [4]. Recently, deep neural networks have achieved remarkable success in several fileds such as image classification and speech recognition. Convolutional neural network (CNN) and recurrent neural network (RNN) have also achieved great success in audio tagging task. Most audio tagging systems are based on the CNN-GRU network structure [5, 6]. In this paper, we will also use this standard CNN-GRU network architecture as our baseline system.

Current deep neural networks require large and varied datasets in order to provide good performance and generalization. However, manually labelling a dataset is expensive and time-consuming. Websites like Youtube, Freesound, or Flickr host large volumes of user-contributed audio and metadata, and labels can be inferred automatically from the metadata. Nevertheless, these automatically inferred labels might include a substantial level of label noise. The effective use of such a large amount of unverified data is the key to improving the performance of audio tagging systems. One of the most basic ideas is to find out the most convincing samples from unverified data during training. Some researchers select reliable unverified data by iterative iteration [7] or setting loss thresholds [8]. In [9], we propose a staged training strategy to pick the most convincing samples from unverified data. We will improve the staged training strategy through an updatable parameter, accelerating the process of training and making the model more stable.

Since audio tagging is a multi-label classification task, where each audio clip may contain multiple tags, sigmoid is naturally the primary choice of the final activation. However, we observe that most audio clips contain only a small number of audio tags. So in [9], we propose a sigmoid-softmax activation structure for audio tagging. Although it has achieved good scores, this method is not general enough. In daily life, an audio clip is likely to contain a variety of audio tags. It is difficult for softmax activation function to solve real multi-label task. So in this paper, we will use sparsemax [10] (a variant of softmax activation function) to compensate for the shortcomings of softmax activation function in multi-label task.

We evaluate our methods on DCASE 2019 Challenge Task 2: *Audio tagging with noisy labels and minimal supervision* [11]. It provides public dataset [12] with baseline. In the challenge of DCASE 2019, our audio tagging system has won the second place. In this work, we will propose a new multi-activation structure and an improved staged training strategy for unverified data.

The rest of the paper is organized as follows. In Section 2, we introduce our baseline system. Section 3 describes our methods in detail. The experiments and results are presented in Section 4 and Section 5. Finally we draw our conclusions in Section 6.

2. BASELINE SYSTEM

This section will briefly introduce our baseline system and main procedure including data pre-processing, neural network and post processing.

2.1. Data pre-processing

We use log-mel energies as acoustic feature. Each audio sample is divided into frames of 40 ms with 50% overlapping. 80 log mel-band energy features are extracted from the magnitude spectrum of each frame. Then we apply sound activity detection by ignoring the silent frames at the beginning and end of each audio. Since the

The corresponding author is Wei-Qiang Zhang.

This work was supported by the National Natural Science Foundation of China under Grant No. U1836219 and the National Key R&D Program of China.

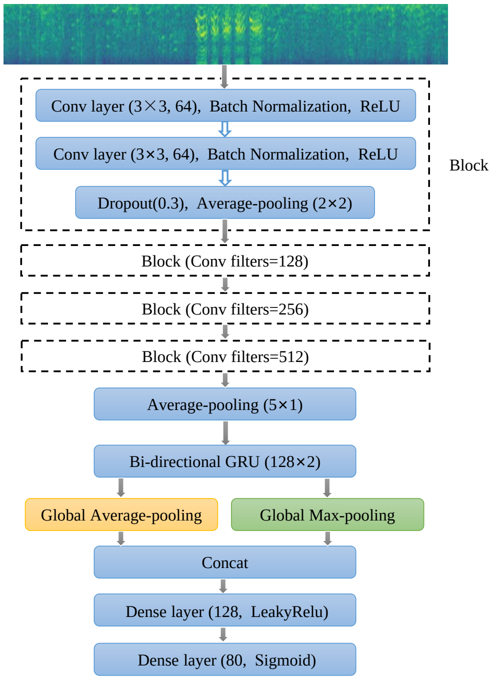
Conv layer (3×3, 64), Batch Normalization, ReLU
Conv layer (3×3, 64), Batch Normalization, ReLU
Dropout(0.3), Average-pooling (2×2)
Block
Block (Conv filters=128)
Block (Conv filters=256)
Block (Conv filters=512)
Average-pooling (5×1)
Bi-directional GRU (128×2)
Global Average-pooling
Global Max-pooling
Concat
Dense layer (128, LeakyRelu)
Dense layer (80, Sigmoid)

Fig. 1. The architecture of CNN-GRU baseline.

length of audio is variable, we fix a target length of 2000 frames and simply repeat the audio clip in case it is too short and downsample to align with the target length in case it is too long. During training, we randomly select continuous 512 frames to feed into the neural network. For test, the whole 2000 frames are used to get predictions.

Meanwhile, we adopt mixup [13] and SpecAugment [14] for data augmentation. In mixup, we randomly select a pair of samples from training data. SpecAugment is implemented by time warping, frequency and time masking. More details are available in [9].

2.2. Neural network

Our neural network is based on CNN-GRU. The network consists of four main parts: four convolutional blocks, one bidirectional GRU, pooling function on time axis and two fully connected layers. The specific network parameters are shown in Figure 1.

2.3. Post processing

The neural network will output probabilities for each class. We normalize the prediction scores to zero mean and unit variance and set min and max zoom to keep the scores between 0 and 1. To enhance system performance, we ensemble our models using geometric average.

3. METHODS

3.1. Staged training strategy for noisy label

In our task, the training set includes a small amount of manually labeled data, whose labels are verified to be correct, and a large amount of unverified data, whose labels may be either correct or incorrect. Since a large amount of data contain noisy label, we try to leverage as much information as possible from unverified data with correct labels and avoid the effects from incorrect labels. Our motivation is that if the model is able to correctly classify the majority of audio clips, the loss from incorrect labels will be larger than the loss from correct labels. So we assume that the sample with large loss during training is more likely to have incorrect label.

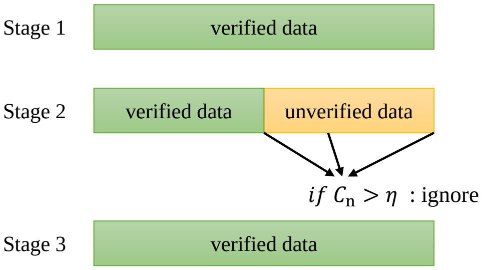

Fig. 2. The illustration of staged training strategy.

In our prior work [9], we propose a staged training strategy to learn from unverified label. First, we only use the verified data to train a preliminary model. Then, we use both the verified and unverified data for training. However, in order to ignore incorrect labels, we adopt a loss masking to ignore the noisy samples with the top k loss in a batch. Finally, we abandon the unverified data and finetune our model with only the verified data.

In [9], k samples are ignored in every batch, regardless of the proportion of incorrect labels in this particular batch. However, in the actual training process, we have to limit one batch to a small size due to the limitation of memory. Thus, the ratio of incorrect labels among batches may vary drastically, leading to varied consequences of this strategy. More specifically, this strategy may ignore many correct labels when the ratio of incorrect labels is relatively low, or keep many incorrect labels when the ratio is high.

So we need to ensure that the selection of ignored data will not be affected by the distribution of incorrect labels in a specific batch. We set an updatable threshold η to decide which sample's loss will be ignored. And η is learnt from the historic top k loss. The specific selection mechanism is defined as follows:

$$L = \sum_i (1 - M_i V_i)\, C_i \tag{1}$$

$$M_i = \begin{cases} 1 & if\ \ C_i > \eta \\ 0 & otherwise \end{cases} \tag{2}$$

$$\eta_t = \alpha\, \eta_{t-1} + (1 - \alpha)\, T_t \tag{3}$$

where C_i is the loss from a single sample in a batch and L is the total loss of the whole batch. M_i equals to 1 if the loss of the i-th sample is greater than η, and otherwise 0. V_i is 1 if the i-th data has verified label and 0 if not. η_t is the current threshold and η_{t-1} is the threshold of last iteration. T_t is the value of ranked k loss in current batch. The setting of k is decided by the noisy label ratio and batch size. α is a smoothing coefficient. In our experiment k is 10 and α is 0.9. The illustration of the staged training strategy is shown in Figure 2. Compared with the mechanism that ignores the noisy samples with the top k loss, the updatable threshold makes model more stable and helps train the network faster.

3.2. Multi-activation structure

Audio tagging is a multi-label classification task, so the general activation of output layer is sigmoid. However, in FSDKaggle2019

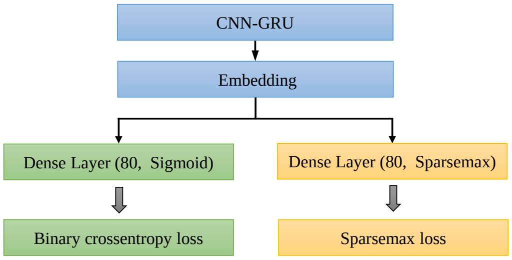

Fig. 3. The illustration of multi-activation structure.

dataset, 84.1% samples in training data have single label. The average number of positive labels is 1.2, which is very close to 1. So we call it a sparse multi-label classification problem. We propose a new structure named sigmoid-softmax activation which combines the advantages of both sigmoid and softmax in [9], which has greatly improved system performance.

But softmax can only solve single-label classification, making the multi-activation structure partially unreasonable. To tackle with the problem, the final activation function needs to be able to solve multi-label classification tasks. On the other hand, its output requires sparse posterior distributions. To meet above requirements, we adopt sparsemax transformation [10] to make model more general and enhance system performance. Sparsemax is a variant of softmax. Softmax function is defined as:

$$\operatorname{softmax}_i(z) = \frac{\exp(z_i)}{\sum_{j=1}^{n} \exp(z_j)} \tag{4}$$

where z_i is the i-th class output score of sample z, and n is the number of output units. In this activation, even very small z_i also outputs a non-zero probability. To get prediction scores with sparse distributions, too small values in softmax should be truncated. Let $\Delta^{K-1} := \left\{p \in \mathbb{R}^K | 1^\top p = 1,\ p \geq 0\right\}$ be the $(K-1)$-dimensional simplex. Sparsemax function is defined as:

$$sparsemax(z) := \underset{p \in \Delta^{K-1}}{\arg\min} \|p - z\|^2 \tag{5}$$

Sparsemax returns the euclidean projection of the input vector z onto the probability simplex. This projection is likely to hit the boundary of the simplex and the output of sparsemax transformation will be sparse. Sparsemax is similar to a truncated version of softmax. Details of the proof and derivations are available in [10]. Accordingly, the loss function would be changed to sparsemax loss which is calculated as follows:

$$z_k = sorted(z), k = 1, 2, \cdots, K \tag{6}$$

$$k(z) = \max\left\{k \in [K] | 1 + k z_{(k)} > \sum\nolimits_{j \leq k} z_{(j)}\right\} \tag{7}$$

$$\tau(z) = \frac{\left(\sum_{j \leq k(z)} z_{(j)}\right) - 1}{k(z)} \tag{8}$$

$$L_{sparsemax}(z; q) = -q^\top z + \frac{1}{2} \sum_{j \in S(z)} (z_j^2 - \tau^2(z)) + \frac{1}{2} \|q\|^2 \tag{9}$$

where the output of neural network z is descendingly sorted as $z_{(1)} \geq z_{(2)} \geq \cdots \geq z_{(K)}$. The target of sparsemax loss is a probability distribution $q \in \Delta^{K-1}$. In experiments, we regard multiple labels as uniformly distributed. Since sparsemax has the distinctive feature, it can return sparse posterior distributions. This property makes it workable to predict multiple labels.

We replace softmax activation function in sigmoid-softmax structure with sparsemax. As shown in Figure 3, one dense layer with sigmoid activation function will be optimized with binary crossentropy loss, and the other dense layer with sparsemax activation function will be optimized with sparsemax loss. The outputs of both dense layers are ensembled to get final prediction.

4. EXPERIMENTS

4.1. Dataset

We demonstrate our proposed methods on the dataset called FSDKaggle2019 [15], which employs audio clips from Freesound Dataset (FSD) [16] and Yahoo Flickr Creative Commons 100M dataset (YFCC) [17]. It consists of 29266 audio recording data, where 24785 recordings are for training and 4481 are for evaluation. Note that only 4970 training data labels are verified manually from FSD, and 19815 recordings with the unverified label come from YFCC. In other words, some labels may suffer from errors. Labels in FSDKaggle2019 are provided at the clip-level, and indicate the presence of a sound category in the audio clip. Audio clips have variable lengths (roughly from 0.3 to 30s).

4.2. Experiment setup

We split training dataset into four folds and apply 4-fold cross validation. Batch size is 64. Adam [18] is used for optimization and the learning rate is 0.001. In the first stage of staged training strategy, all data come from verified dataset and run for 10k iterations. In the second stage, the proportion of verified dataset is equal to unverified dataset and run for 10k iterations. In the third stage, only verified dataset is used and run for 5k iterations.

4.3. Metric

The primary competition metric is label-weighted label-ranking average precision (lwlrap) . This measures the average precision of retrieving a ranked list of relevant labels for each test clip. The label-weighted means that the overall score is the average over all the labels in the test set, where each label receives equal weight. Lwlrap is the macro-average of per-class LRAP and LRAP is calculated as:

$$\operatorname{LRAP}(y, \hat{f}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \frac{1}{\|y_i\|_0} \sum_{j: y_{ij}=1} \frac{|\mathcal{L}_{ij}|}{\operatorname{rank}_{ij}} \tag{10}$$

where $\mathcal{L}_{ij} = \left\{k : y_{ik} = 1, \hat{f}_{ik} \geq \hat{f}_{ij}\right\}$, $\operatorname{rank}_{ij} = \left|\left\{k : \hat{f}_{ik} \geq \hat{f}_{ij}\right\}\right|$, $|\cdot|$ computes the cardinality of the set, and $\|\cdot\|_0$ computes the number of nonzero elements in a vector.

Besides, during model training, we also evaluate the classification performance of each class by F-score. F-score is the harmonic average of precision and recall.

5. RESULTS AND ANALYSIS

5.1. Experimental results

The lwlrap scores on both cross-fold validation and private evaluation dataset are shown in Table 1. Systems with multi-activation

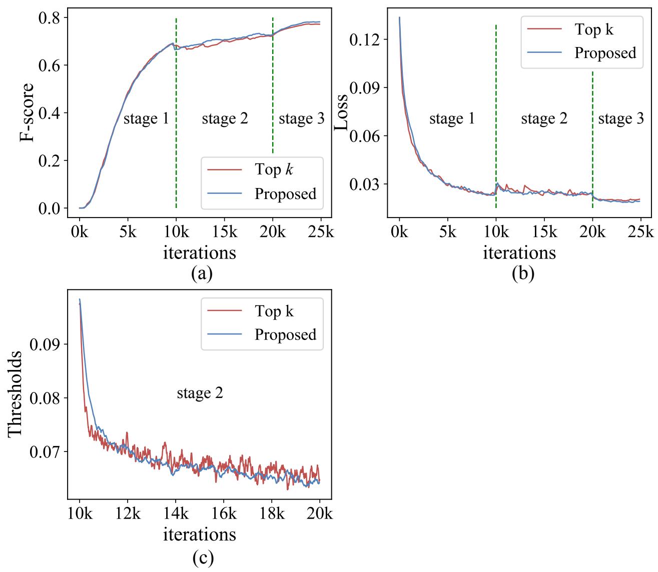

Fig. 4. The comparison between the top k selection mechanism [9] and our proposed updatable parameter selection mechanism. (a) is the variation of F-score among iterations. (b) is the variation of loss. (c) is the variation of selection threshold.

Table 1. Lwlrap scores on both cross-fold validation and private evaluation dataset. The score on cross-fold validation dataset are the average of scores on four folds. "sig-soft" is the abbreviation of sigmoid-softmax activation structure and "sig-sparse" is the abbreviation of sigmoid-sparsemax activation structure. Top k represent the top k selection mechanism [9] and "proposed" represent updatable parameter selection mechanism.

		Verified data	Verified and unverified data	
			top k	proposed
Cross-fold Validation	sigmoid	0.8417	0.8512	0.8569
	sig-soft	0.8376	0.8561	0.8617
	sig-sparse	0.8423	0.8602	0.8631
Private Evaluation	sigmoid	0.7155	0.7253	0.7278
	sig-soft	0.7207	0.7388	0.7402
	sig-sparse	0.7236	0.7372	0.7411

structure outperform systems with single activation. Performance of sigmoid-sparsemax is slightly better than sigmoid-softmax on both cross-fold validation and private evaluation dataset. Sigmoid-sparsemax structure is used to compensate for the shortcomings of softmax in multi-label task. In order to testify its performance, we evaluate the lwlrap scores of samples with multi labels on cross-fold validation. As shown in Table 2, sigmoid-sparsemax structure can improve the performance on data with multi labels. Compared with only using verified data, staged training strategy can learn from noisy label to improve performance. As for selection mechanisms in the staged training strategy, performance of the updatable parameter selection mechanism outperforms top k selection mechanism.

We compare the performance of the top k selection mechanism [9] and our proposed updatable parameter selection mechanism during training in Figure 4. Subfigure (a) shows the variation of F-score among iterations. (b) shows the variation of training loss. (c) shows the variation of selection threshold. In the first 10k iterations, the training is at the first stage and we use only verified data for training. From 10k to 20k iterations, the training is at the second stage, where different selection mechanisms are adopted. From 20k to 25k iterations, the training is at the third stage and only verified data are used for fine-tuning. In the first stage, the performances of two mechanisms are almost the same, because the same data and same training strategy are used. In the second stage, proposed mechanism outperforms top k mechanism in terms of F-score, training loss. Besides, as shown in subfigure (c), the selected threshold of proposed mechanism is more stable than top k selection mechanism. In the third stage, proposed mechanism also achieves higher F-score and lower loss than top k selection mechanism.

Table 2. Samples with multi labels lwlrap scores on cross-fold validation.

	fold1	fold2	fold3	fold4	average
sig-soft	0.7617	0.8097	0.7941	0.7643	0.7825
sig-sparse	0.7658	0.8154	0.7865	0.7964	0.7910

Table 3. Comparison of several systems, on both public leaderboard and private leaderboard. (1) OUmed: DCASE 1st place model; (2) Ebbers: DCASE 3rd place model; (3) THUEE: DCASE 2nd place model.

	Lwlrap (public LB)	Lwlrap (private LB)
THUEE [9]	0.7423	0.7575
Proposed	***	**0.7591**
OUmed [19]	**0.7474**	0.7579
Ebbers [20]	0.7305	0.7552

5.2. Comparison with other methods

As shown in Table 3, compared with other state-of-the-art methods, the performance of our model is competitive. Public leaderboard is used for system development during Kaggle competition session, and private leaderboard is used for final evaluation after all submissions. As this Kaggle competition has closed, we currently have no access to the groundtruth labels of public leaderboard dataset, so we use cross-fold validation for development, and private leaderboard for evaluation. Note that model esemble is used in all systems in Table 3. To enhance system performance, we ensemble our models using geometric average.

6. CONCLUSION

In this paper, we propose a staged training strategy and a sigmoid-sparsemax multi-activation structure to tackle the task of sparse multi-label audio tagging using training data with noisy label. This paper is an improvement and extension of our prior work for participation in Task 2 of DCASE 2019 Challenge. We replace previous top k selection mechanism in staged training strategy with a updatable parameter selection mechanism, which makes the selection of ignored noisy labelled data more stable and reasonable. Besides, we also substitute previous sigmoid-softmax activation structure with sigmoid-sparsemax activation to fit the setting of multi-label classification. Our methods have shown state-of-the-art performance on audio tagging task.

7. REFERENCES

[1] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *IEEE Conference on Advanced Video & Signal Based Surveillance*, 2007.

[2] Sacha Krstulović, "Audio event recognition in the smart home," in *Computational Analysis of Sound Scenes and Events*, pp. 335–371. Springer, 2018.

[3] Dan Stowell and David Clayton, "Acoustic event detection for multiple overlapping similar sources," in *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2015, pp. 1–5.

[4] Anurag Kumar and Bhiksha Raj, "Audio event detection using weakly labeled data," in *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 2016, pp. 1038–1047.

[5] Yong Xu, Qiuqiang Kong, Qiang Huang, Wenwu Wang, and Mark D Plumbley, "Attention and localization based on a deep convolutional recurrent model for weakly supervised audio tagging," *arXiv preprint arXiv:1703.06052*, 2017.

[6] Turab Iqbal, Qiuqiang Kong, Mark D Plumbley, and Wenwu Wang, "General-purpose audio tagging from noisy labels using convolutional neural networks," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018*. Tampere University of Technology, 2018, pp. 212–216.

[7] Matthias Dorfer and Gerhard Widmer, "Training general purpose audio tagging networks with noisy labels and iterative self-verification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, 2018.

[8] Il-Young Jeong and Hyungui Lim, "Audio tagging system for dcase 2018: focusing on label noise data augmentation and its efficient learning," *Tech. Rep., DCASE Challenge*, 2018.

[9] Kexin He, Yuhan Shen, and Weiqiang Zhang, "Multiple neural networks with ensemble method for audio tagging with noisy labels and minimal supervision," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (2019)*, June 2019, to appear.

[10] Andre Martins and Ramon Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *International Conference on Machine Learning*, 2016, pp. 1614–1623.

[11] Eduardo Fonseca, Manoj Plakal, Frederic Font, Daniel PW Ellis, and Xavier Serra, "Audio tagging with noisy labels and minimal supervision," *arXiv preprint arXiv:1906.02975*, 2019.

[12] Eduardo Fonseca, Jordi Pons Puig, Xavier Favory, Frederic Font Corbera, Dmitry Bogdanov, Andres Ferraro, Sergio Oramas, Alastair Porter, and Xavier Serra, "Freesound datasets: a platform for the creation of open audio datasets," in *Hu X, Cunningham SJ, Turnbull D, Duan Z, editors. Proceedings of the 18th ISMIR Conference; 2017 oct 23-27; Suzhou, China.[Canada]: International Society for Music Information Retrieval; 2017. p. 486-93.* International Society for Music Information Retrieval (ISMIR), 2017.

[13] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[14] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[15] Eduardo Fonseca, Manoj Plakal, Frederic Font, Daniel P. W. Ellis, and Xavier Serra, "Audio tagging with noisy labels and minimal supervision," in *Submitted to DCASE2019 Workshop*, NY, USA, 2019.

[16] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[17] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li, "Yfcc100m: The new data in multimedia research," *arXiv preprint arXiv:1503.01817*, 2015.

[18] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] Osamu Akiyama and Junya Sato, "Multitask learning and semi-supervised learning with noisy data for audio tagging," Tech. Rep., DCASE2019 Challenge, June 2019.

[20] Janek Ebbers and Reinhold Haeb-Umbach, "Convolutional recurrent neural network and data augmentation for audio tagging with noisy labels and minimal supervision," Tech. Rep., DCASE2019 Challenge, June 2019.