ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/330535184

Iterative Knowledge Distillation In R-CNNs for Weakly-Labeled Semi-Supervised Sound Event Detection

Conference Paper · November 2018

CITATIONS
8

READS
161

3 authors, including:

Khaled Koutini
Johannes Kepler University Linz
11 PUBLICATIONS **9** CITATIONS

SEE PROFILE

Gerhard Widmer
Johannes Kepler University Linz
377 PUBLICATIONS **7,693** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Combinbing Expert Knowledge with Deep Learning Methods for Audio Analysis and Event Detection View project

Project Musical Harmony Analysis Using Artificial Neural Networks View project

All content following this page was uploaded by Khaled Koutini on 22 January 2019.

The user has requested enhancement of the downloaded file.

ITERATIVE KNOWLEDGE DISTILLATION IN R-CNNS FOR WEAKLY-LABELED SEMI-SUPERVISED SOUND EVENT DETECTION

Khaled Koutini, Hamid Eghbal-zadeh, Gerhard Widmer

Institute of Computational Perception (CP-JKU),
Johannes Kepler University Linz, Austria
khaled.koutini@jku.at

ABSTRACT

In this paper, we present our approach used for the CP-JKU submission in Task 4 of the DCASE-2018 Challenge. We propose a novel iterative knowledge distillation technique for weakly-labeled semi-supervised event detection using neural networks, specifically Recurrent Convolutional Neural Networks (R-CNNs). R-CNNs are used to tag the unlabeled data and predict strong labels. Further, we use the R-CNN strong pseudo-labels on the training datasets and train new models after applying label-smoothing techniques on the strong pseudo-labels. Our proposed approach significantly improved the performance of the baseline, achieving the event-based f-measure of 40.86% compared to 15.11% event-based f-measure of the baseline in the provided test set from the development dataset.

Index Terms— Weakly-labeled, Semi-supervised, Knowledge Distillation, Recurrent Neural Network, Convolutional Neural Network

1. INTRODUCTION

Motivated by the release of Audioset [1], the task of predicting strong labels using models trained on weakly-labeled audio data was introduced in the DCASE-2017 challenge (task 4) [2]. However, in DCASE-2018, the task has changed and transformed into a semi-supervised task which adds another dimension of complexity to this challenge. By leaving the majority of the training data unlabeled [3], the organizers motivated the participants to leverage the large sets of unlabeled data in a semi-supervised manner in order to improve the performance of their systems. Another important change compared to DCASE-2017 is the evaluation metric, that is changed from segment-based evaluation to event based evaluation. In DCASE-2018 task4, the submissions will be evaluated by the macro average of class-wise *event-based* F1-scores (explained in Section 4.3). The new evaluation metric introduces new challenges to the task, since the systems need to predict the onsets and offsets of the events very accurately. In other word, unlike DCASE-2017, events that are partially detected – with inaccurate onsets and offsets– do not improve the performance based on the new evaluation metric, but rather worsen it, as it will get evaluated as both a false positive and a false negative [3]. In this paper, we propose a novel approach to overcome the difficulties of this new task by leveraging the unlabeled data via an iterative knowledge distillation in neural networks. We show that using our method, the performance of a Convolutional Recurrent Neural Network (R-CNN) can be significanlt improved. We provide experimental results on DCASE-2018 task 4 dataset and compare it with the baselines we used. The remainder of the paper is as follows. Section 2 describes the related work. In Section 3 we explain our proposed method. The experiments and the empirical results are presented in Section 4 and finally Section 5 concludes the paper.

2. RELATED WORK

2.1. Weakly Labels Sound event detection

To deal with weak labels, it is important to pay attention to the power of state-of-the-art tagging systems. By using a R-CNN architecture, Xu et al. [4] achieved the best tagging performance in DCASE 2017 task4. Their architecture uses gated activations of convolutional and recurrent layers and an attention mechanism to locate the events. Their architecture consists of multiple gated convolutional layers followed by a bi-directional Gated Recurrent Unit (GRU). Between convolutional blocks, they used max-pooling only on the frequency dimension, in order to keep the time information required for event localization.

Lee et al. [5] used an ensemble of multiple deep convolution neural networks trained on audio clips of different lengths and managed to achieve the best event detection performance in DCASE 2017 task 4. They showed the power of an ensemble model for such tasks, following an ensemble method proposed by Caruana et al. [6] by iteratively adding models that increase the performance of the whole system.

2.2. Knowledge Distillation In Neural Networks

A considerable amount of work has been done in transferring the knowledge between models either for compressing models while maintaining their performance [7, 8, 9, 10] or for increasing the interpretability and explaining the decisions [11, 12, 13]. A pioneer idea of knowledge transfer from a large model or an ensemble of multiple models to a simple model was introduced by Bucila et al. [7] in the context of compressing large models into small models that are more suitable for deployment. Ba and Caruana [8] empirically showed that a similar performance to the state-of-the-art deep neural network models can be achieved using shallow models. This performance of shallow models can not be achieved by training on the original training data, but rather by training shallow model to mimic the output activations of a deep model. Further work by Hinton et al. [9] showed that these simple models (also known as student model) can even perform better than the models they mimic, by distilling the knowledge from an ensemble of deeper models (known as teacher models) into a single new model (the student). They managed to improve the performance of their models, both on the MNIST dataset [14] and for an Automatic Speech Recognition task (ASR).

Furlanello et al. [10] managed to make student models surprisingly outperform their teacher models on many computer vision and language modeling tasks, by retraining the student models with identical parameterization to their teachers, but with different initialization. They trained the student models to predict the correct labels, and further to match the output distribution of the teacher.

3. THE PROPOSED APPROACH

In this section, we detail the key components of our proposed iterative knowledge distillation method.

3.1. Key Differences With Previous Work

We adopted a deep architecture (described in Table 2) inspired by the one proposed by Xu et al. [4]. However, We used the ReLu activation function for the convolutional layers and kept the gated linear activation only for the recurrent layer. we used the Simple Recurrent Unit (SRU) [15] as a recurrent unit because of its fast training. We achieved empirically a better tagging performance using a global average of the frame level probabilities, instead of the attention mechanism proposed by Xu et al.. Our shallow model (Table 3) is inspired by the DCASE-2018 task 4 baseline model [3] with an adjusment of replacing the recurrent unit in the baseline system with an SRU.

Unlike the approaches stated in Section 2.2, we trained our new student models on the smoothed predictions over the time-dimension of the teacher models. We show that smoothing is an important step given the nature of our task. Namely, we used *median smoothing* with varying window size and with/without Gaussian filter smoothing (Figure 1). We repeated this step a couple of times, although the improvement was diminishing over the steps. We also trained deep and shallow models in each iteration, as follows. We used the predictions of the best model for each class as the pseudo-labels of the next iteration for knowledge distillation. In comparison with Furlanello et al. [10], they trained the new models with the supervision of only the latest iteration of a single model, while Hinton [9], Bucila [7] and their collaborators used an ensemble of teacher models in a non-iterative manner. And finally, we used the smoothed labels, while the aforementioned methods use the probabilities of the teacher to train the students.

3.2. Proposed Approach for Audio Tagging

We train an R-CNN on the weakly-labeled dataset and predicted pseudo-weak-labels for both in-domain and out-of-domain sets. Table 2 shows the configuration of the layers of the model.

3.3. The Proposed Approach for Strong Label Prediction

We follow a multi-pass strategy to get our final predictions, by iteratively predicting pseudo-strong-labels for the labeled, in-domain and out-of-domain sets, and retraining new models on those new predictions.

3.3.1. The First Pass

We trained a recurrent convolutional neural network with the same architecture that was used for tagging (Table 2). However, the network is not only trained on the provided labels of the labeled set, but also on the predicted pseudo labels for both the in-domain and out-of-domain sets. The result of the first pass are strong labels for the labeled, in-domain and out-of-domain sets. These labels are presented in the form of frame-level probabilities for every audio clip.

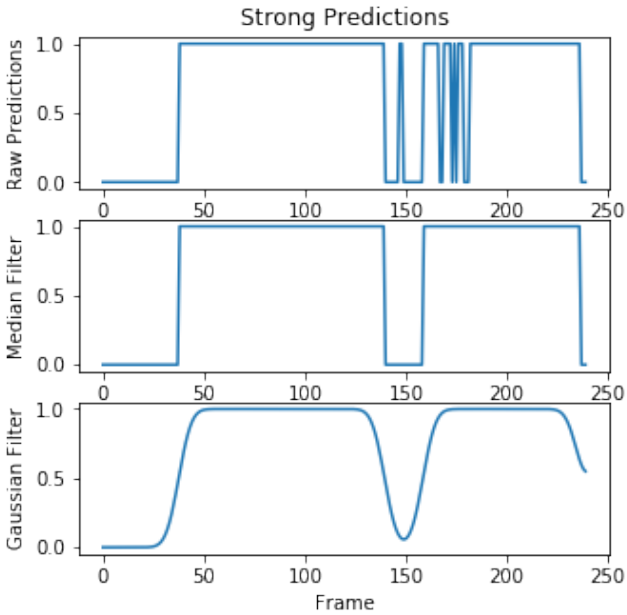

Figure 1: Example of strong predictions before/after smoothing.

3.3.2. The Second Pass

In the second pass, we smooth the current predicted pseudo-strong labels using median/Gaussian filters and we train new models on them. We observed that the performance of the models varies among different classes. We achieved better performances in some classes using a deep model (Table 2), while for other classes shallow models (Table 3) performed better. In addition, using median smoothing with or without Gaussian smoothing resulted in varying performances for different classes.

3.3.3. Model selection

We train multiple models with/without smoothing. Then, we select the best trained model for each class to predict new pseudo-strong-labels for the respected class for the labeled, in-domain and out-of-domain sets. Using these new prediction, we iteratively repeated the second pass (Figure 2).

3.3.4. Smoothing for Strong Prediction

The strong predictions of our models trained only on weakly-labeled data tend to be noisy. Therefore, we smooth those predictions using median and Gaussian filters (Figure 1). We then use these smoothed probabilities for retraining the network in the next pass as explained in Section 3.3.

Table 1: F-score results per class by pass. The average is calculated class-wise (macro-average) [16]. SM: shallow model (Table 3). DM: Deep model (Table 2). Merge: merging models by taking the prediction of the best model for each class. Note that for pass 1 (marked with *) the models are trained using weak-labels. From pass 2 onwards, the models are trained on the smoothed strong predictions of the previous pass.

Pass	Config.	Average	Alarm	Blender	Cat	Dishes	Dog	Electric..	Frying	Runnin..	Speech	Vacuum..
1*	SM	17.43	7.1	12.3	2.2	4.9	4.8	38.7	36.6	13.0	2.5	52.1
1*	DM	24.65	26.0	24.3	26.3	13.2	23.5	33.3	12.4	16.2	41.8	29.4
2	SM	35.18	39.8	33.3	32.1	15.8	25.3	40.0	38.8	22.5	47.3	56.9
2	DM	34.08	33.9	31.9	32.1	15.4	24.9	39.3	37.1	22.5	47.3	56.3
3	SM	34.46	41.8	32.5	33.3	16.1	16.3	40.0	40.0	22.4	47.1	55.1
3	DM	33.59	37.3	33.8	30.0	15.6	20.5	35.7	36.6	20.9	48.3	57.1
4	SM	34.46	41.8	32.5	33.3	16.1	16.3	40.0	40.0	22.4	47.1	55.1
4	DM	34.67	42.5	32.8	32.6	14.0	21.4	42.9	37.7	20.9	46.9	54.9
5	SM	35.48	43.9	38.3	31.1	13.1	21.5	40.6	41.3	22.1	48.2	54.7
5	DM	34.85	40.4	39.5	33.5	14.5	20.6	42.1	39.3	18.1	46.4	54.0
6	Merge	37.89	46.6	38.2	45.6	14.2	24.2	41.3	40.0	28.0	47.6	53.3
7	Merge	39.11	46.0	39.5	41.1	18.1	25.7	43.3	43.1	28.4	48.7	57.1
8	Merge	40.86	49.3	40.0	50.0	18.1	25.7	44.1	43.5	31.0	49.9	57.1

Table 2: Proposed deep architecture for predicting strong labels and audio tagging. BN: Batch normalization, BIAS: Model uses bias with no batch normalization, ReLu: Rectified Linear activation function

Input 240×64	
2×2 Conv(pad-1, stride-1)-64-BN-ReLu 2×2 Conv(pad-1, stride-1)-64-BN-ReLu 1×2 Max-Pooling	
2×2 Conv(pad-1, stride-1)-64-BN-ReLu 2×2 Conv(pad-1, stride-1)-64-BN-ReLu 1×2 Max-Pooling	
2×2 Conv(pad-1, stride-1)-64-BN-ReLu 2×2 Conv(pad-1, stride-1)-64-BN-ReLu 1×2 Max-Pooling	
2×2 Conv(pad-1, stride-1)-64-BN-ReLu 2×2 Conv(pad-1, stride-1)-64-BN-ReLu 1×2 Max-Pooling	
1×1 Conv(pad-1, stride-1)-256-BIAS-ReLu 1×4 Max-Pooling	
Bi-directional SRU 128 hidden units	
1×1 Conv(pad-1, stride-1)-10-BIAS-Sigmoid Output 240×10	
(Strong predictions) Output 240×10	(Weak-label training and tagging) Global-Average-Pooling Output 10

Table 3: Proposed shallow architecture for predicting strong labels. Similar to the baseline [3]. BN: Batch normalization, BIAS: Model uses bias with no batch normalization, ReLu: Rectified Linear activation function

Input 240×64	
3×3 Conv(pad-1, stride-1)-64-BN-ReLu 1×4 Max-Pooling	
3×3 Conv(pad-1, stride-1)-64-BN-ReLu 1×4 Max-Pooling	
3×3 Conv(pad-1, stride-1)-64-BN-ReLu 1×4 Max-Pooling	
Bi-directional SRU 128 hidden units	
1×1 Conv(pad-1, stride-1)-10-BIAS-Sigmoid Output 240×10	
(Strong predictions) Output 240×10	(Weak-label training and tagging) Global-Average-Pooling Output 10

4. EXPERIMENTS AND RESULTS

4.1. Dataset

The dataset is split into a training set, a test set and an evaluation set [3]. The training set contains three subsets, a labeled set, an unlabeled-in-domain set and an unlabeled-out-of-domain set. In this paper, they are referred to as labeled, in-domain, out-of-domain respectively. The test set contains 288 strongly labeled audio clips. The evaluation set consist of 880 audio clips, for which our system predicted strong labels for the challenge submission.

4.2. Features Extraction

We use log-scaled Mel-bands spectrograms as an input for all our models. We extracted 64 Mel bands from 64 ms frames with 22.5 ms overlap using Librosa [17]. That resulted in an input size of 240 $\times$ 64 for our models.

4.3. Evaluation Metric

The evaluation metric for the task is the event-based F-score [16]. The predicted events are compared with a reference event list, by comparing the onset and the offset of the predicted event with the overlapping reference event. The predicted event is considered correctly detected (true positive), if it's onset is within 200 ms collar of the reference event onset and its offset is within 200 ms or 20% of the event length collar around the reference offset. If a reference event has no matching predicted event, it is considered a false negative. If the predicted event doesn't match any reference event, it is considered a false positive. Furthermore, if the system partially predicted an event without accurately detecting its onset and offset,

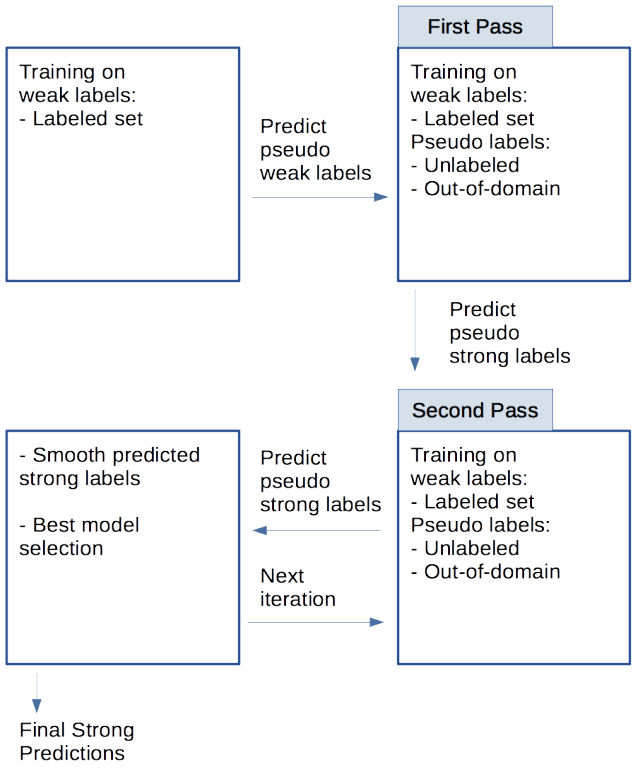

Figure 2: The proposed knowledge distillation framework for RCNNs.

it will be penalized twice, as a false positive and a false negative. Equation (1) shows the calculation of the F-score for each class [3].

$$F_c = \frac{2.TP_c}{2.TP_c + FP_c + FN_c}, \quad (1)$$

Where F_c, TP_c, FP_c, FN_c are the F-score, true positives, false positives, false negatives of the class c respectively. The final evaluation metric the average of the F-score for all the classes.

4.4. Results

Table 1 shows the class-wise intermediate results over the training iterations. In the first pass, the shallow and the deep models where trained on the given weak labels and on the predicted pseudo weak labels for the in-domain and out-of-domain sets. We show that the shallow model works better only for the classes Electric shaver/toothbrush, frying and vacuum cleaner. We justify that by the nature of these classes, as they tend to be longer, with the event-length medians 8.78, 10.00, 9.99 respectively, compared to 1.03 the event-length median for all the classes (Table 1 in [3]). Therefore, we conclude that the shallow model fails to localize when trained on weak labels. However, the shallow models works surprisingly well when trained on the strong prediction of the previous pass. They even generalize better in many cases then the deep models (passes 2 to 5 for many classes). By merging the predictions of the best model for each class iteratively, we managed to push the performance of the system to 40.86%.

Table 4 shows the final macro-averaged event-based evaluation results on the test set compared to the baseline system.

Table 4: The performance of our approach compared to the baseline system [3]. Note that we re-ran the baseline on our machines, hence the slight difference from the reported values in [3].

	F1	Precision	Recall
Baseline	15.11	14.20	17.80
Our system	40.86	40.21	44.42

5. CONCLUSION

In this paper we propose a method for detecting sound events from weakly-labeled data. We proposed iteratively training similar models with different initialization on the smoothed predictions of the previous iteration. The goal behind this is to iteratively making the detected sound events more precise and predicting the onsets and the offsets of the events more accurately. We provide empirical evidence that this iterative process makes the predicted time boundaries for individual events more accurate, in accordance with the results of [10]. The event-based F-score increases over iterations to reach 40.86% on the test set, compared to the baseline performance of 15.11%. We also show empirically that shallow models trained on the predictions of deep models can even generalize better then their teachers, in line with the results of [8].

6. ACKNOWLEDGMENT

This work has been supported by (1) the COMET-K2 Center for Symbiotic Mechatronics of the Linz Center of Mechatronics (LCM) and (2) the COMET Center SCCH, with funds provided by the Austrian Federal Government, the Federal State of Upper Austria, and the Austrian Ministries BMVIT and BMWFW.

7. REFERENCES

[1] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 776–780.

[2] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

[3] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. Parag Shah, "Large-Scale Weakly Labeled Semi-Supervised Sound Event Detection in Domestic Environments," July 2018, submitted to DCASE2018 Workshop. [Online]. Available: https://hal.inria.fr/hal-01850270

[4] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," *arXiv preprint arXiv:1710.00343*, 2017.

[5] D. Lee, S. Lee, Y. Han, and K. Lee, "Ensemble of convolutional neural networks for weakly-supervised sound event detection using multiple scale input," Tech. Rep., DCASE2017 Challenge, Tech. Rep., 2017.

[6] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, 2004. [Online]. Available: http://doi.acm.org/10.1145/1015330.1015432

[7] C. Bucila, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, 2006, pp. 535–541. [Online]. Available: http://doi.acm.org/10.1145/1150402.1150464

[8] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 2654–2662. [Online]. Available: http://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep

[9] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[10] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born-again neural networks," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018, pp. 1602–1611. [Online]. Available: http://proceedings.mlr.press/v80/furlanello18a.html

[11] R. Chandra, K. Chaudhary, and A. Kumar, "The combination and comparison of neural networks with decision trees for wine classification."

[12] S. Tan, R. Caruana, G. Hooker, and A. Gordo, "Transparent model distillation," *arXiv preprint arXiv:1801.08640*, 2018.

[13] N. Frosst and G. E. Hinton, "Distilling a neural network into a soft decision tree," in *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017 co-located with 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017), Bari, Italy, November 16th and 17th, 2017.*, 2017. [Online]. Available: http://ceur-ws.org/Vol-2071/CExAIIA_2017_paper_3.pdf

[14] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Nov 2012.

[15] T. Lei and Y. Zhang, "Training rnns as fast as cnns," *arXiv preprint arXiv:1709.02755*, 2017.

[16] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[17] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015, pp. 18–25.

View publication stats