SOUND EVENT DETECTION USING WEAKLY LABELED SEMI-SUPERVISED DATA WITH GCRNNS, VAT AND SELF-ADAPTIVE LABEL REFINEMENT

Robert Harb

Graz University of Technology, Austria
robert.harb@student.tugraz.at

Franz Pernkopf

Graz University of Technology, Austria
Signal Processing and Speech Communication Laboratory
pernkopf@tugraz.at

ABSTRACT

In this paper, we present a gated convolutional recurrent neural network based approach to solve task 4, large-scale weakly labeled semi-supervised sound event detection in domestic environments, of the DCASE 2018 challenge. Gated linear units and a temporal attention layer are used to predict the onset and offset of sound events in 10s long audio clips. Whereby for training only weakly-labeled data is used. Virtual adverserial training is used for regularization, utilizing both labelled and unlabelled data. Furthermore, we introduce self-adaptive label refinement, a method which allows unsupervised adaption of our trained system to increase the quality of frame-level class predictions. The proposed system reaches an overall macro averaged event-based F-score of 34.6%, resulting in a relative improvement of 20.5% over the baseline system.

***Index Terms*—** DCASE 2018, Convolutional neural networks, Sound event detection, Weakly-supervised learning, Semi-supervised learning

1. INTRODUCTION

In this paper we summarize the methods we use to solve task 4 of the DCASE 2018 challenge, the large-scale weakly labeled semi-supervised sound event detection in domestic environments.

The proposed method uses a gated convolutional recurrent neural network (GCRNN). This is similar to the best model [1] of last years DCASE 2017 challenge task 4 [2] which also used a GCRNN based approach. As an extension to the attention mechanism we introduce an algorithm we call self-adaptive label refinement, which uses unlabeled input data and clip-level class predictions to refine the frame-level predictions of our model. To incorporate the provided unlabeled data we use virtual adverserial training (VAT) [3]. VAT has, amongst others, already been used successfully in semi-supervised text [4], image classification [3] tasks and acoustic event detection [5]. Furthermore VAT showed competitive performance against other deep semi-supervised learning algorithms [6].

2. PROPOSED METHOD

2.1. Gated convolutional recurrent neural network

The winning team of last year's DCASE SED (sound event detection) task [1] showed that using gated linear units (GLUs) [7] instead of commonly used activation functions like rectified linear units (RELUs) or leaky ReLUs in the CRNN is an eligible approach for SED.

Gating mechanisms have been used successfully in a variety of neural network architectures. For example in RNNs using LSTM [8] cells, which have a separate input, output and forget gate. The rough idea behind gating mechanisms is to have a gate which can control how information flows in the network.

In the setting of SED, the GLU units should adapt their behavior such that they act as an attention mechanism on the time-frequency (T-F) bin of each feature map. They can set their value value close to one if information related to any of the considered audio events passes through, and otherwise block the flow of unrelated information by setting their value close to zero.
GLUs are defined as follows:

$$\mathbf{Y} = (\mathbf{W} * \mathbf{X} + \mathbf{b}) \odot \sigma(\mathbf{V} * \mathbf{X} + \mathbf{c}), \tag{1}$$

Where $\mathbf{W}$ and $\mathbf{V}$ denote the convolutional filters with their respective biases $\mathbf{b}$ and $\mathbf{c}$, σ is the sigmoid function, $\mathbf{X}$ denotes the input to the layer, and $\odot$ denotes elementwise multiplication.

Figure 1 shows how the gated CNN blocks are incorporated into the network, whereby in our model we use three subsequent gated CNN blocks.

2.2. Virtual adverserial training

We make use of (VAT) [3] for regularization. The virtual adverserial loss is defined such that the robustness of the model's posterior distribution $p(\mathbf{y}|\mathbf{x})$ is increased for small and bounded perturbations of the input $\mathbf{x}$.

The adversarial perturbation $\mathbf{r}_{v\text{-}adv}$ is computed by maximizing a non-negative distance function between the unperturbed $p(\mathbf{y}|\mathbf{x};\hat{\theta})$ and perturbed $p(\mathbf{y}|\mathbf{x}+\mathbf{r};\theta)$ posterior. Whereby $\hat{\theta}$ denotes the current model parameter.

When using VAT the following additional cost is added to the objective function:

$$\mathrm{KL}[p(\mathbf{y}|\mathbf{x};\hat{\theta})||p(\mathbf{y}|\mathbf{x}+\mathbf{r}_{\mathrm{v\text{-}adv}};\theta)]. \tag{2}$$

The Kullback-Leibler divergence KL is used as distance function between $p(\mathbf{y}|\mathbf{x};\hat{\theta})$ and $p(\mathbf{y}|\mathbf{x}+\mathbf{r};\theta)$, and $||\mathbf{r}||$ is limited to the sphere around $\mathbf{x}$ with some radius $\leq \epsilon$.

$$\mathbf{r}_{v\text{-}adv} = \arg\max_{\mathbf{r},\|\mathbf{r}\|\leq\epsilon} \mathrm{KL}(p(\mathbf{y}|\mathbf{x};\hat{\theta})||p(\mathbf{y}|\mathbf{x}+\mathbf{r};\theta)) \tag{3}$$

Since the virtual adversarial perturbation only requires input $\mathbf{x}$ and does not require label $\mathbf{y}$, VAT is applicable to semi-supervised training.

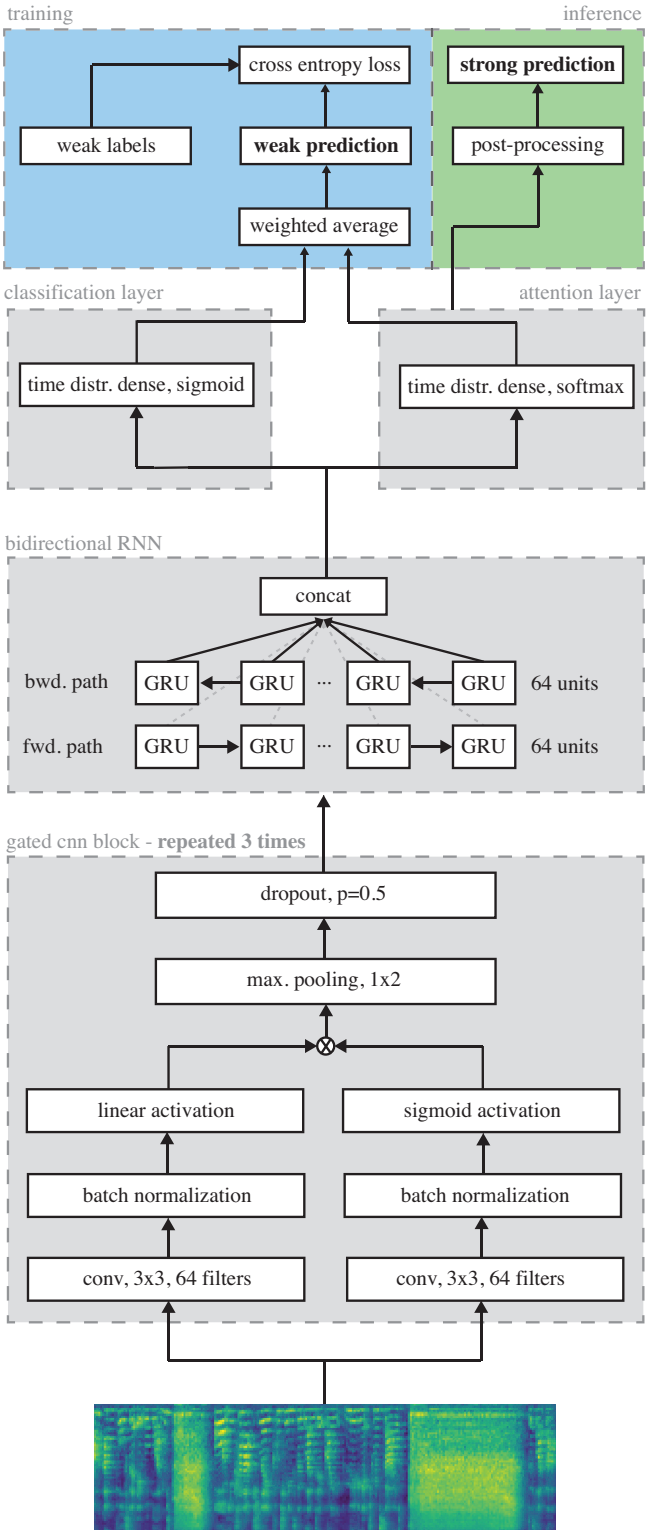

Figure 1: Network structure

2.3. Attention mechanism

To predict the temporal locations of each audio event which is presented in a given input sample, we use a similar approach as used in [1]. We extend it by using a method based on weak and strong prediction alignment to select for each event class an appropiate post-processing. Whereby the term weak prediction is used to refer to predictions at clip-level and strong prediction is used to refer to class predictions at time-level.

As depicted in Figure 1, the output of a bidirectional RNN is fed into both an attention and a classification layer. The classification layer uses a sigmoid activation function to predict the probability of each occurring class at each timestep. While the attention layer uses a softmax activation over all classes. Intuitively, using a softmax in the attention layer should aid the network to learn to pick the most dominant class at each frame. Although this might not be an ideal approach if temporal overlaps of multiple events are occuring, since then a more dominant event might be able to suppress the activation of another one.

The final prediction o for the weak labels is determined by the weighted average of the element-wise multiplication of the attention and classification layer output:

$$o = \frac{\sum_t^T \mathbf{z}_{cla}(t) \odot \mathbf{z}_{att}(t)}{\sum_t^T \mathbf{z}_{att}(t)}, \tag{4}$$

Where $\mathbf{z}_{cla}$(t) and $\mathbf{z}_{att}$(t) are the outputs of the classification layer and of the attention layer. T denotes the frame-level resolution of the input spectogram.

Figure 2 shows the output of the classification and attention layer for one audio clip of the development set containing several events labeled as dog. It can be seen that there is a clear correlation between ground truth event labels and the activations of the attention and classification layer. However it is not obvious how to extract the exact start and end points of each individual event from the layer activations. Our experiments showed that just taking the product of the attention and classification layer activations, thresholded with a fixed value for all classes, e.g. 0.5, gives unsatisfactory results. Also it has been shown in similar weakly labeled SED settings that the trained network adapts differently for different classes [9]. Especially there seems to be a difference between classes which tend to have short event durations in contrast to classes who span the majority of timesteps of a clip. Considering this, it might be necessary to use a different post-processing for each class to account for that.

The fact that no strong event annotations are available for training makes this a non-trivial problem, otherwise a simple approach would be to test several post-processing methods and select for each class the one which gave best performance.

We introduce self-adaptive label refinement, where we check the alignment between strong and weak predictions, and use this as an approximate prediction how well a given post-processing method performs at extracting the right onset and offset of events. Using this approach we can use unlabelled data to estimate how well a given post-processing parameterization performs for each class, and take the best performing parameterization for our final strong prediction.

For post-processing we threshold the output value of the classification layer, followed by a median filter. Therefore the parameters we vary in each iteration are the threshold, and the width of the median filter.

In particular, when training has finished, the following steps are repeated on each class:

1. A full forward pass is performed to create weak and strong predictions for each clip. Whereby for each class, strong predictions are only considered if the respective weak prediction is positive.
2. For each detected event in each clip, a new sample is created containing only the frames of the clips original spectogram where the detected event occurs according to the strong prediction. Those new samples which possibly contain the class, are labelled as 1.

   Additionally each time events of a class are detected in a clip, another new sample is created which contains only the temporal frames of the original spectogram where no occurence of the given class was predicted. Those are all labelled to 0.
3. The generated new samples are then passed through the network. Using the resulting weak predictions and the labels set beforehand, a crossentropy loss for each class is calculated. This loss indicates how good the weak and strong predictions align.
4. For each class the post-processing with the smallest loss value is selected.

This approach does not need any labels, neither strong nor weak. Therefore our method for post-processing selection is applicable using data of both, the weakly-supervised and the unsupervised dataset. Also the method can be used to adapt the post-processing at inference time to new unseen data.

2.4. Training

For each sample the cross entropy loss is calculated between the predicted probabilities for each class and the weak ground truth labelling:

$$E = -\frac{1}{N}\sum_{i}^{N}\sum_{c}^{M} l_c^{(i)} log(y_c^{(i)}), \tag{5}$$

Where the number of classes is denoted by M, the number of weakly labeled 10 second audio clips by N, $y_c^{(i)}$ denotes the predicted probability for class c of sample i, and $l_c^{(i)}$ is the given binary label in the weakly labelled test set.

In each step a batch containing an equal distribution of samples from the labelled and unlabelled data set is processed. The total loss consists of the cross entropy loss of the labelled samples, regularized with VAT depending on both the labelled and unlabelled samples weighted by a factor λ:

$$\begin{aligned} L = &-\frac{1}{N}\sum_{i,c}^{N,M} l_c^{(i)} log(y_c^{(i)}) \\ &+ \lambda \sum_{i}^{N+K} \mathrm{KL}(p(\mathbf{y}|\mathbf{x}^{(i)};\hat{\theta})||p(\mathbf{y}|\mathbf{x}^{(i)}+\mathbf{r};\theta)), \end{aligned} \tag{6}$$

Where K denotes the number of unlabelled in-domain audio clips. We did not use any of the provided out-of-domain data.

The loss was optimized using Adam [10] with a learning rate of 0.001 and a batch size of 30. The network was implemented using tensorflow [11].

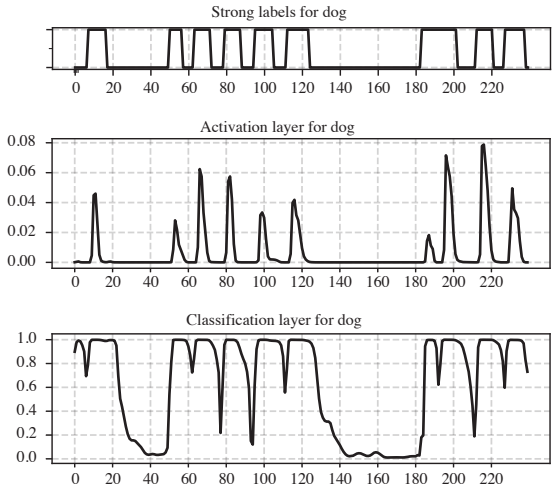

Figure 2: Classification and attention layer activations for file: *Y0a8RB5eOGJ4_30.000_40.000.wav* and class dog

3. EXPERIMENTS AND RESULTS

3.1. Dataset

The method is evaluated using a subset of the Google Audioset [12], which was provided with task 4 of the DCASE 2018 challenge[13].

All audioclips are of 10-second length and contain one or multiple sound events of 10 different classes. Whereby different events may overlap. The dataset consists of a training, testing and evaluation subset.

The training subset consists of 1,578 weakly labeled clips, an unlabeled in-domain set of 14,412 clips and an unlabeled out-of-domain set of 39,999 clips extracted from classes that are not considered in task 4.

The test set contains 288 clips, whereby the distribution in terms of clips per class is similar to the weakly labeled training set. For the test set strong labels from human annotators are given, therefore timestamps for the onset and offset of each event in the clip are included. For training only weak labels are used. The weak labels indicates if a given event occurs somewhere in a 10s audio clip, however no information about the onset and offset of the events, nor how often the event occurs is given. This setting can also be considered as a multiple instance learning (MIL) problem [9].

Log-Mel filter banks are used as features. Each sample is split into 240 frames by 64 mel frequency channels. Before the filters are calculated, each sample is converted to a mono signal with a sampling rate of 44,100Hz.

1. A model is trained with the available weak labels. The trained first model is then used to predict labels for the unlabeled in domain set.
2. A second model is then trained on the unlabeled in domain set predictions of the first model, and the weakly labeled set is used to validate the model.

	baseline		no adaption		adaption to train set		adaption to test set	
	F1	**ER**	**F1**	**ER**	**F1**	**ER**	**F1**	**ER**
Alarm_bell_ringing	-	-	27.9%	1.38	21.0%	1.14	18.2%	1.12
Blender	-	-	27.9%	1.52	23.2%	1.33	38.1%	0.97
Cat	-	-	29.9%	2.87	19.2%	1.54	25.2%	1.30
Dishes	-	-	4.9%	1.93	32.5%	1.16	32.5%	1.16
Dog	-	-	29.3%	2.00	2.3%	1.36	15.8%	1.36
Electric_shaver_toothbrush	-	-	7.4%	2.61	40.0%	0.96	40.0%	0.96
Frying	-	-	14.1%	3.79	40.0%	1.50	40.7%	1.46
Running_water	-	-	18.0%	1.89	31.1%	1.22	32.4%	1.21
Speech	-	-	22.6%	1.25	41.3%	0.97	40.2%	0.98
Vacuum_cleaner	-	-	37.5%	2.58	40.5%	1.31	63.0%	0.75
Baseline	14.06%	1.54	21.8%	2.18	29.1%	1.25	**34.6%**	**1.1**

Table 1: **Class-wise results** on the **development set**, total scores are macro averaged

3.2. Evaluation

For evaluation the macro averaged event-based F-score [14] is used. The event-based metrics are calculated using an open source toolbox called sed_eval [15]. As given by the dcase challenge, for calculation of event-based metrics a 200ms collar on onsets and a 200ms / 20% of the events length collar on offsets was set. For calculation of the total performance over all individual classes, macro averaging is used. This has the effect that each class has equal influence on the final metrics, even if the distribution of classes in the tested set is unbalanced.

3.3. Results

Table 1 shows the event based F1 scores and error rates of our system on the development set. Whereby we compare the resulting scores when we did no refinement, and when we performed self-adaptive label refinement using data of the training and development set.

For the post-processing of the system with no refinement, we used a fixed threshold of 0.5 for all classes and no median filter on the output.

All three systems perform better than the baseline system. Whereby using self-adaptive label refinement gives a significant performance increase, whereby the increase is bigger when the adaption was done on the development set.

4. CONCLUSION

In this paper, we proposed a method for sound event detection using only weakly labeled and unsupervised data. Our approach is based on GCRNNs, whereby we introduce self-adaptive label refinement, a method which can be used to adapt the model to unlabelled data, and increase SED performance.

Our final system performance is with 34.6% significantly higher than the score of the baseline system 14.06%.

5. REFERENCES

[1] 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.

[2] http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/index.

[3] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning," *ArXiv e-prints*, Apr. 2017.

[4] T. Miayto, A. M. Dai, and I. Goodfellow, "Virtual adversarial training for semi-supervised text classification," 2016.

[5] M. Zhrer and F. Pernkopf, "Virtual adversarial training and data augmentation for acoustic event detection with gated recurrent neural networks," pp. 493–497, 08 2017.

[6] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and I. J. Goodfellow, "Realistic evaluation of semi-supervised learning algorithms," 2018. [Online]. Available: https://openreview.net/forum?id=ByCZsFyPf

[7] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *CoRR*, vol. abs/1612.08083, 2016. [Online]. Available: http://arxiv.org/abs/1612.08083

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[9] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *CoRR*, vol. abs/1804.10070, 2018.

[10] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 12 2014.

[11] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: a system for large-scale machine learning."

[12] 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.

[13] http://dcase.community/workshop2018/.

[14] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016. [Online]. Available: http://www.mdpi.com/2076-3417/6/6/162

[15] A. Heittola, T.; Mesaros, "sed_eval - evaluation toolbox for sound event detection." https://github.com/TUT-ARG/sed_eval, accessed: 2018-07-20.