Time Aggregation Operators for Multi-label Audio Event Detection

Pankaj Joshi, Digvijaysingh Gautam, Ganesh Ramakrishnan, Preethi Jyothi

Dept. of Computer Science and Engineering, Indian Institute of Technology Bombay

{panjoshi,digvijay,ganesh,pjyothi}@cse.iitb.ac.in

Abstract

In this paper, we present a state-of-the-art system for audio event detection. The labels on the training (and evaluation) data specify the set of events occurring in each audio clip, but neither the time spans nor the order in which they occur. Specifically, our task of weakly supervised learning is the "Detection and Classification of Acoustic Scenes and Events (DCASE) 2017" challenge [5]. We use the winning entry in this challenge given by Xu *et al.* [10] as our starting point and identify several important modifications that allow us to improve on their results significantly. Our techniques pertain to aggregation and consolidation over time and frequency signals over a (temporal) sequence before decoding the labels. In general, our work is also relevant to other tasks involving learning from weak labeling of sequential data.

1. Introduction

Multi-label audio event detection is a task that comprises of detecting various audio classes present in the audio file. Machine learning (ML) models can be trained using some labeled data to do this audio class detection. However, in practice, the available labeling may be weak, that is, the class labels might not be available with associated timestamps of the audio events. A task of learning to predicting these labels under weak supervision was presented as the Audio tagging task in the *Detection and Classification of Acoustic Scenes and Events* (DCASE) 2017 challenge [5]. The challenge was to evaluate systems for the large-scale detection of sound events using weakly labeled training data. The best performing system in this challenge by Xu *et. al.* [10] models the temporal structure of each sound clip using a recurrent neural network. It also benefits in robustness and numerical stability by making use of gated convolutional units in the earlier ReLU activations.

1.1. Related Work

Adavanne *et al.* [1] and Parascandolo *et al.* [7] propose the first uses of convolutional architectures for sound event detection using weakly labeled data. They benefited from the use of 2-d convolutions over audio spectrograms presumably because the audio class is influenced by interpolations on both the time and frequency domains. As mentioned above, the winner [10] of the challenge replaced ReLU activations with a gated unit comprising of a linear and sigmoid. This is done to introduce attention to each layer of the neural network. They get additional benefits by employing a recurrent neural network later to model the temporal structure of each sound clip.

While most approaches have implicitly associated the clip-level labels with every segment in it, some like Yu *et al.* [11], Feng *et al.* [3] and Tseng *et al.* [9] have viewed a clip as a set of instances, where each instance is a fixed image/audio segment and approached the problem as a multi-instance, multi-labeled (MIML) problem. However, this treatment did not yield the best reported results.

Xu *et al.* [10] also predicted the time span for the audio activity using a soft argmax operator (which they refer to as softmax based attention).

1.2. Our Contributions

- We build upon the best performing system to advance state-of-the-art (weakly supervised learning approach) on the DCASE 2017 sound event detection task.
- We show that a suite of simple operators that aggregate evidence across time can further improve the performance significantly. These operators aggregate a sequence of time-indexed vectors into a single vector. In particular, we consider a "max-of-means" operator which averages the vectors in a sliding window across time and aggregates the resulting vectors using a coordinate-wise maximum operation.
- We demonstrate significant benefits of the proposed operators using extensive experimentation and also present ablation tests to draw further insights.

We believe that our proposed operators can complement existing best practices for many audio and video classification tasks.

2. Model architecture

Our architecture is inspired by the publicly available code of Xu *et al.* [10] which is the winning entry of the challenge. This architecture used the Gated Linear Unit (GLU) of Dauphin *et al.* [2] in place of the Relu activation by Nair *et al.* [6]. GLUs use a sigmoid gate, the value of which determines the flow of information. A GLU can be defined as:

$$Y = (W * X + b) \otimes \sigma(V * X + c)$$

Here σ is the sigmoid operator whereas $\otimes$ is the element wise multiplication and X is the input. We use 4 Gated Convolution blocks, each block containing two gated convolutions followed by an operator that consolidates signals from both **t**ime and **f**requency units and is therefore referred to as a *t-f* operator. Each convolution block contains two convolution layers with 64 filters of size 3×3 with gated linear units as activation. Our suite of *t-f* operators includes max pool over frequency, max pool over time, mean over time as well as some of their compositions such as max across frequency with mean across time, max pool across both frequency and time, *etc.*

Following the four gated convolutional blocks described earlier is another convolution layer of 256 filters of size 3×3 followed by another max pool across frequency. The output from this convolutional network is fed into a BiRNN block with 128 GRU units and GLUs in place of activation. The output from the previous layer is fed into fully connected layers with softmax and sigmoid activations. The weights in the softmax layer are used to compute a weighted mean of the sigmoid layer

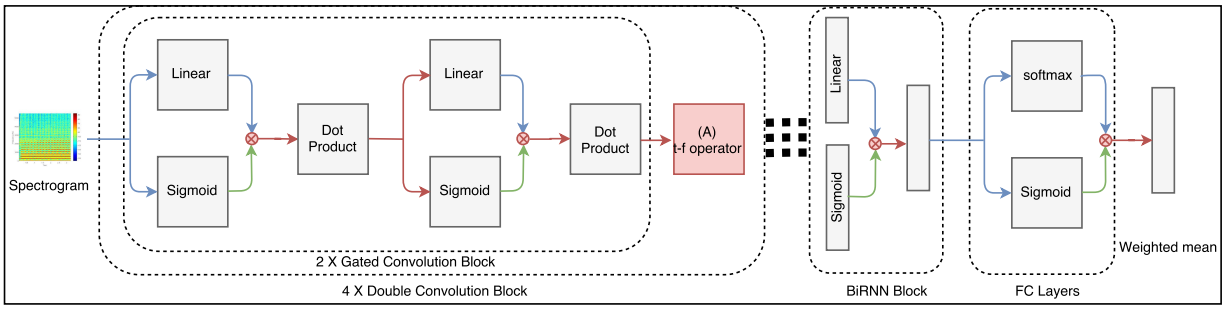

Figure 1: *Network Architecture. (A) t-f operators*

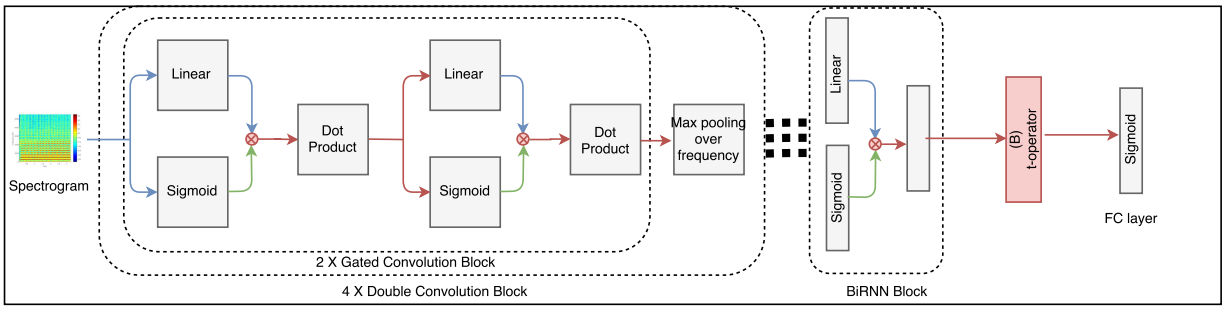

Figure 2: *Network Architecture. (B) t-operator*

to finally get the label predictions. The softmax can also be used to predict time stamps for sound events in the audio clip. For this, we take cue from the work of Xu *et al.* [10], and check if jointly training a model to classify the audio segment while also learning to predict the time-stamp using a soft-argmax operator mutually help each other. While we find marginal gains using this joint learning, we translate this somewhat insightful experiment into another alternative (the mean over time).

In the first variant of the model, following the BiRNN layer, we use an operator that consolidates signals on the time unit, hereafter referred to as a *t*-operator. The suite of *t*-operators include mean over time [8], a similar max over time and their compositions such as max over mean over time. In this case, we use max over frequency as *t-f* operator. The output from the *t*-operator is fed into a fully connected layer with sigmoid unit that is trained to predict labels.

In another variant of this architecture, instead the Gated Convolution blocks, we use 5 convolutional blocks, wherein each block contains two convolutional layers with same number of filters followed by a max over frequency. The number of filters is gradually increased in each block from 16 to 256. In this architecture, we use ReLU as our activation function for the convolutional blocks. This layer is followed by BiRNN layer and the rest of the architecture is same as the architecture for the *t*-operator. We call this model ReLU convolutions model. We suggest one more variant for the ReLU convolutions model where we replace the Gated Linear units in the BiRNN block with ReLU we call it ReLU CNRNN model.

In Figure 1, we present our network architecture highlighting the place of *t-f* operators. Whereas, in Figure 2, we remove the fully connected block at the end of the previous network and instead feed output from the BiRNN block to one of our proposed *t*-operators.

3. Experiments

3.1. Dataset

For our experiments, we used the training, development and evaluation sets predefined by DCASE 2017 for the audio event detection challenge [5]. These datasets are fixed subsets of the Google AudioSet corpus [4] and consist of sound clips from 17 different audio events relating to vehicle sounds ('Car', 'Train', *etc.*) and warning sounds ('Fire engine', 'Ambulance siren', *etc.*). The training, development and evaluation sets contain a total of 51172, 488 and 1103 audio clips, respectively. These datasets are weakly labeled in that only the types of the audio events occurring in the clip are provided, but the timestamp of the events are not specified. In each of these datasets, the label distribution across the 17 audio event classes is highly unbalanced. For example, the 'Car' class is most frequent with a total of 25, 744 clips whereas the 'Car alarm' class is least frequent with a total of 273 clips.

3.2. Implementation Details

Log-mel filterbank features extracted from the raw audio waveforms were fed as input to our architecture. Each training clip has 240 time steps with 64 mel frequency bands in each time step. Our model implementation largely borrows from the publicly available code of the winning system [10]. We adopt three training strategies that were initially proposed for the winning system: 1) Batches were created such that even the least frequent classes were included in each batch. 2) Instead of setting a fixed threshold across all classes, class-specific thresholds were determined by tuning on the development set. 3) Predictions across epochs for the same system were averaged in order to improve its stability.

Model	F1 Score	Precision	Recall
State of the art [10]	0.567	0.538	0.601
State of the art (Fusion) [10]	0.577	0.565	0.589
Max across both (Benchmark)	0.570	0.531	0.614
Max Frequency	0.575	0.540	0.614
Avg Time Max Freq	0.582	0.545	0.625
Avg Freq	0.527	0.487	0.574

Table 1: *Comparison of models on development Set for different t-f operators*

Model	F1 Score	Precision	Recall
State of the art [10]	0.542	0.589	0.502
State of the art (Fusion) [10]	0.556	0.614	0.508
Max across both (Benchmark)	0.570	0.534	0.611
Max Freq	0.546	0.506	0.592
Avg Time Max Freq	0.557	0.516	0.605
Avg Freq	0.519	0.473	0.574

Table 2: *Comparison of models on evaluation Set for different t-f operators*

3.3. Results

We first evaluate the utility of our t-f operators which appear before the BiRNN block in our architecture. Tables 1 and 2 show the F-1 score/precision/recall breakup for different t-f operators. The first two rows correspond to the current state-of-the-art scores for this task, as reported in Xu et al. [10], from their best-performing individual system and fusion system, respectively. The third row corresponds to our extension of the best individual system from Xu *et. al.* [10]. Our reimplementation leads to slightly higher numbers on both the development and evaluation sets; this system is henceforth referred to as "Benchmark". Each of the t-f operators that we have suggested, with the exception of mean over frequency, outperforms the "benchmark" on the development set but not always on the evaluation set. We believe this could be due to differences in the distribution of sound events across the development and evaluation sets: Since we tune threshold values on the development set, this difference in class distribution could likely affect performance on the evaluation set.

Tables 3 and 4 show results from using our t-operators following the BiRNN block in our architecture. For easy comparison, we show the state-of-the-art scores from Xu et al. [10] again in both the tables. On both the development and evaluation sets, we observe consistent improvements for every choice of our t-operator. The "mean over time" t-operator improves even over the "state of the art (Fusion)" system shown in Tables 1 and 2. The "max over time" operator improves only marginally compared to the "mean over time" operator on the development set (i.e. 0.599 vs. 0.591), but provides a larger improvement on the evaluation set (0.575 vs. 0.561). The "max over mean" operator was designed to combine the best properties of both the preceding operators: Average a fixed set of time-indexed vectors in a sliding window across time and use a coordinate-wise maximum operation to reduce it to a single vector. This is the best-performing t-operator both on the development and evaluation sets.

In Tables we experiment with the "ReLU convolutions" and "ReLU CNRNN" variants of the convolutional network that we

Model	F1 Score	Precision	Recall
State of the art [10]	0.567	0.538	0.601
State of the art (Fusion) [10]	0.577	0.565	0.589
Mean over time	0.591	0.556	0.630
Max over time	0.599	0.574	0.625
Max over mean over time	0.601	0.590	0.614

Table 3: *Comparison of models on the development set for different t-operators*

Model	F1 Score	Precision	Recall
State of the art [10]	0.542	0.589	0.502
State of the art (Fusion) [10]	0.556	0.614	0.508
Mean over time	0.561	0.517	0.615
Max over time	0.575	0.541	0.613
Max over mean over time	0.590	0.567	0.616

Table 4: *Comparison of models on the evaluation set for different t-operators*

described earlier in Section 2. We used the "max over mean" t-operator (which performed the best, as shown in Tables 3 and 4) with both these variants. We find additional benefits in performance from using these two variants with the t-operator. To assess the importance of the t-operator, we also experimented with using the "ReLU CNRNN" networks without adding the t-operator after the BiRNN layers; this did not perform as well, the results are reported in Table 5 and 6.

3.4. Class-wise Analysis

In Tables 7 and 8, we summarize the class-wise F1 scores of all the different models on the development set and the evaluation set, respectively. For each class, the best F1 score is highlighted in bold. The column "Net wins" lists the net number of classes on which each model outperforms the benchmark model.

From the tables, we observe the following:

- Firstly, we observe that *all* our systems have a positive

Model	F1 Score	Precision	Recall
ReLU Convolution	0.610	0.573	0.652
ReLU CNRNN (no t-operator)	0.601	0.565	0.642
ReLU CNRNN ((2,2) pool)	0.589	0.583	0.594
ReLU CNRNN (with t-operator)	0.601	0.576	0.629

Table 5: *Performance of the ReLU based variants on the development set*

Model	F1 Score	Precision	Recall
ReLU Convolution	0.576	0.535	0.623
ReLU CNRNN (no t-operator)	0.582	0.538	0.634
ReLU CNRNN ((2,2) pool)	0.574	0.561	0.589
ReLU CNRNN (with t-operator)	0.591	0.562	0.624

Table 6: *Performance of the ReLU based variants on the evaluation set*

Table 7: *Class-wise comparison of models on development set*

Model	Train horn	Air horn, truck horn	Car alarm	Reversing beeps	Bi-cycle	Skate-board	Ambu-lance	Fire engine	Civil def-ense siren	Police car	Scream-ing	Car	Car pass-ing by	Bus	Truck	Motor-cycle	Train	Net wins
Max across both (Benchmark)	0.79	0.53	0.62	0.67	0.51	0.76	0.55	0.61	0.82	**0.63**	0.64	0.40	0.36	0.51	0.44	0.52	0.69	0
t-operator: mean over time	0.84	0.63	0.55	0.76	0.54	0.70	0.56	0.59	0.84	0.61	0.69	0.48	0.42	0.45	0.50	0.58	0.70	+7
t-operator: max over time	**0.87**	0.55	0.64	**0.77**	0.52	0.76	0.53	0.56	0.81	0.60	0.72	0.48	**0.50**	0.49	0.51	**0.59**	0.67	+4
t-operator: max after mean	0.81	**0.64**	0.71	0.71	0.51	0.72	0.60	0.60	0.79	0.55	0.77	**0.52**	0.42	0.43	0.48	0.56	0.72	+6
ReLU CNRNN (no t-operator)	0.84	0.58	**0.79**	0.73	0.58	0.76	0.50	0.53	0.79	**0.63**	0.71	0.50	0.36	0.48	0.52	**0.59**	0.71	+6
ReLU CNRNN (with t-operator)	0.83	0.53	0.69	0.69	**0.60**	**0.78**	0.48	**0.65**	0.83	**0.63**	0.71	0.49	0.38	0.44	0.53	0.50	**0.73**	+9
ReLU CNRNN ((2,2) pool)	0.77	0.57	0.57	0.71	0.53	0.72	0.55	0.53	0.80	0.60	**0.78**	0.44	0.36	**0.55**	**0.55**	0.56	0.69	+2
ReLU Convolution (with t-operator)	0.86	0.56	0.73	0.76	0.56	0.74	**0.62**	0.58	**0.85**	0.62	**0.78**	0.51	0.39	0.50	0.50	0.52	**0.73**	+8

Table 8: *Class-wise comparison of models on evaluation set*

Model	Train horn	Air horn, truck horn	Car alarm	Reversing beeps	Bi-cycle	Skate-board	Ambu-lance	Fire engine	Civil def-ense siren	Police car	Scream-ing	Car	Car pass-ing by	Bus	Truck	Motor-cycle	Train	Net wins
Max across both (Benchmark)	0.63	0.52	0.51	0.37	0.37	0.57	**0.67**	0.59	**0.85**	0.57	0.77	0.62	**0.31**	0.34	**0.50**	0.60	0.73	0
t-operator: mean over time	0.62	0.53	0.64	0.42	0.37	**0.66**	0.62	0.60	0.80	0.36	0.82	0.62	0.27	0.42	0.44	0.59	0.74	+1
t-operator: max over time	0.63	0.58	0.62	0.40	0.39	0.64	0.58	**0.61**	0.84	0.48	**0.84**	0.64	0.27	0.37	0.48	**0.65**	0.71	+4
t-operator: max after mean	**0.65**	0.58	**0.65**	0.40	0.36	0.64	0.57	0.59	0.80	0.52	0.81	0.64	0.27	0.39	0.48	**0.65**	0.76	+4
ReLU CNRNN (no t-operator)	0.60	0.57	0.59	0.47	0.46	0.62	0.56	0.48	0.77	**0.64**	0.79	**0.65**	0.26	0.43	0.49	0.58	0.74	+3
ReLU CNRNN (with t-operator)	0.63	**0.60**	0.59	**0.48**	**0.48**	0.60	0.57	0.60	**0.85**	0.53	0.80	**0.65**	0.25	0.41	0.47	0.64	0.76	+7
ReLU CNRNN ((2,2) pool)	0.61	0.58	0.55	0.44	0.42	0.62	0.60	0.53	0.84	0.43	0.79	0.64	0.28	0.40	0.48	0.53	**0.79**	+1
ReLU Convolution (with t-operator)	0.63	0.57	0.60	0.44	0.43	0.59	0.52	0.59	0.84	0.55	0.81	0.64	0.30	**0.45**	0.49	0.57	0.74	+3

score for net wins (both on the development set and the evaluation set).

- In terms of net wins, the ReLU CNRNN model performs the best.
- The t-operator provides consistent gains in performance. This can be clearly seen, for example, when we compare the net wins of ReLU CNRNN (no t-operator) and ReLU CNRNN (with t-operator) on both the development and evaluation sets.
- For most of our models, the evaluation set shows somewhat less improvements compared to the development set. (Indeed, for 4 classes, the benchmark model performs the best in the evaluation set, whereas this happens for only one class (with a tie) in the development set.) This can be attributed to the fact that some of the classes are sparsely represented in the data and hence the corresponding results can be sometimes inconsistent for those specific classes and overall show high variance.
- Different classes have different systems performing well on them. Overall, we observe that the classes from the group of "Warning" sounds (i.e. Train horn, Air horn, Truck horn, Car alarm, Reversing beeps, Ambulance, Police Car, Fire engine, Civil defense siren and Screaming) perform better than the classes from the group of "Vehicle sounds" (i.e. all classes except ones that are categorized as "Warning sounds") across all models. This is generally true for humans as well, in that humans find it easier to accurately identify sirens as opposed to vehicle sounds.

"Car passing by" is an example of a class that performs poorly on the evaluation set across all models. This poor performance could also be attributed to the quality of the annotations for this particular class of sounds. The Audioset [4] website lists a quality value for each audio class by conducting an internal quality assessment of a random sample of videos for each class. "Car passing by" was not estimated to be of high quality; only 5/10 (i.e. 50%) of the random samples were found to be accurate for this class. This high amount of noise in the labels is likely to have contributed to this class performing poorly across all the models.

4. Conclusion and Future Work

In summary, we present a set of simple time-aggregation operators that provide significant improvements on an audio detection task where we surpass the previously best-performing system to set a new state-of-the-art. We provide a detailed analysis of the performance of our systems across different audio events, which gives us further insights about which events are easier to predict and when we can expect degradation in performance. For future work, we hope to scale our model so that it works efficiently with a larger set of classes from the Audioset corpus [4]. We also intend to explore better strategies to automatically learn good thresholds for each class rather than manually determine class-specific thresholds.

5. References

[1] S. Adavanne, P. Pertilä, and T. Virtanen. Sound event detection using spatial features and convolutional recurrent neural network. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 771–775. IEEE, 2017.

[2] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*, 2016.

[3] J. Feng and Z.-H. Zhou. Deep miml network. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17))*, pages 1884–1890, 2017.

[4] 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*, pages 776–780. IEEE, 2017.

[5] 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.

[6] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[7] G. Parascandolo, T. Heittola, H. Huttunen, T. Virtanen, et al. Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6):1291–1303, 2017.

[8] K. Taghipour and H. T. Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, 2016.

[9] S.-Y. Tseng, J. Li, Y. Wang, J. Szurley, F. Metze, and S. Das. Multiple instance deep learning for weakly supervised audio event detection. *arXiv preprint arXiv:1712.09673*, 2017.

[10] 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.

[11] C. Yu, K. S. Barsim, Q. Kong, and B. Yang. Multi-level attention model for weakly supervised audio classification. *arXiv preprint arXiv:1803.02353*, 2018.