Audio event detection on Google's Audio Set database: Preliminary results using different types of DNNs

Javier Darna Sequeiros and Doroteo T. Toledano

AuDIas – Audio, Data Intelligence and Speech, Universidad Autónoma de Madrid

javier.darna@estudiante.uam.es, doroteo.torre@uam.es

Abstract

This paper focuses on the audio event detection problem, in particular on Google Audio Set, a database published in 2017 whose size and breadth are unprecedented for this problem. In order to explore the possibilities of this dataset, several classifiers based on different types of deep neural networks were designed, implemented and evaluated to check the impact of factors such as the architecture of the network, the number of layers and the codification of the data in the performance of the models. From all the classifiers tested, the LSTM neural network showed the best results with a mean average precision of 0.26652 and a mean recall of 0.30698. This result is particularly relevant since we use the embeddings provided by Google as input to the DNNs, which are sequences of at most 10 feature vectors and therefore limit the sequence modelling capabilities of LSTMs.

Index Terms: Audio Set, deep neural network, audio event recognition, machine learning.

1. Introduction

In machine learning, there are several problems that try to mimic biologic senses, such as recognizing objects in images (image object recognition), or identifying particular sounds in an audio track (audio event recognition). In recent years, there have been great improvements in image object recognition thanks to the availability of large databases such as the Imagenet database [1] and similar ones. These have allowed the organization of competitions and have fostered the proposal of novel network architectures such as the Alexnet, VGG, Residual Networks, etc. In the case of audio event recognition, the lack of availability of large databases has not allowed a similar development until very recently. There have been several notable efforts to foster research in this area, among which we must mention:

- The CLEAR audio event recognition and classification challenge [2], which compared algorithms on a database with 12 audio event classes including common sounds from meeting rooms and seminars.
- The urban sound taxonomy [3] dataset, containing 10 classes of urban sounds.

Since 2013, the *Detection and Classification of Acoustic Scenes and Events* (DCASE) community (http://dcase.community/) has organized several challenges focused on different acoustic scenes and events detection and classification problems. Since 2016, there are yearly challenges and workshops. The 2018 edition [4] received 223 submission entries from 81 teams, which implies a huge collective effort in the area. It proposed five different tasks, one of them being about general-purpose audio tagging. The database used in this task is a subset of the Freesound database [5], a collaborative database of sounds under a Creative Commons license. This subset has 11.1K segments and 41 different categories taken from the Google Audio Set ontology.

Along with these databases and challenges, several models based on neural networks were developed in order to classify the data in such databases, quickly becoming state-of-art over previous models mostly based on hidden Markov models (HMM):

- The paper "Polyphonic sound event detection using multi label deep neural networks" [6] uses a neural network for multi label audio event classification that obtained 63.8% accuracy, outperforming the previous state-of-art model based on HMMs.
- The paper "Recurrent Neural Networks for Polyphonic Sound Event Detection in Real Life Recordings" [7] proposes a bi-directional LSTM network to classify audio events from a database with 61 classes from 10 different contexts. This system reports an average F1-score of 65.5%.

Nevertheless, these studies focus on somewhat restricted tasks, and the databases used are relatively small, in terms of both number of samples and number of classes, which seriously limits the applicability of modern deep learning techniques and the progress made in audio event recognition for general cases.

For those reasons, Google created a database named Google Audio Set consisting of segments of 10 seconds extracted from YouTube videos, and published it in 2017 [8]. With over 2 million samples and 527 different classes, this database is significantly larger and wider than any other database ever created for this problem, thus allowing to train and test more versatile models. In particular, such a huge database is very well fitted to the problem of deep learning where very complex models can be trained from huge amounts of data. In addition to the database itself, Google trained a model on it in order to establish a baseline. This model, described as a multilayer perceptron with a single hidden layer of units (hence not a *deep* neural network) produced a mean average precision (mAP) of 0.314 [9] on the evaluation subset. This value can be used as a baseline for other models, but unfortunately, Google did not publish more details on this system. As the database was published just one year before the moment this article was written, there has not been great improvement from the previously mentioned baseline. Despite that, the current state-of-art results have been achieved through the use of attention models, reaching mAPs of 0.327 with the inclusion of a trainable probability measure for the

10.21437/IberSPEECH.2018-14

samples [10], and 0.360 with the implementation of several levels of attention [11].

This paper intends to be a first approach to Google Audio Set. Our goal is to train several different architectures of neural networks with this database and compare their evaluation results with each other and with Google's baseline. The rest of the paper is organized as follows: Section 2 will discuss in more detail the Google Audio Set database. Section 3 will define the neural networks that were used to face the audio event detection problem, section 4 will describe the tests that were made on the neural networks created, section 5 will interpret the results from those tests, and finally section 6 will conclude the work and propose future research lines that arise from the results of this paper.

2. The Google Audio Set database

The Google Audio Set database is available in two different formats:

- Text files describing the video id, start time, stop time, and labels assigned to each segment.
- Features (embeddings) extracted at 1 Hz for each segment using a DNN trained by Google (the structure of this DNN is similar to the VGG networks used in image recognition).

In this paper we have used only the latter format so that minimal preprocessing is required and results are easier to recreate. However, the official dataset from Google Audio Set webpage was not used because it was developed to be used directly on Tensorflow (also developed by Google) and it was very difficult and inefficient to use in other toolkit such as Keras. Instead, we finally used an “unofficial” conversion to .h5df format available from the Google+ user group "audioset-users" [12]. This conversion includes, for each segment, the extracted audio features as a uniform 128x10 array and the presence or absence of each possible label as a boolean vector.

The Google Audio Set database consist of 3 subsets (in any of the available formats):

- Balanced training, which has a balanced distribution of the classes but contains only a small fraction of the samples (about 22K segments).
- Unbalanced training, which contains all of the samples (2.0M segments) but suffers from a greatly unbalanced distribution of the classes.
- Evaluation, which contains about 20K segments.

In all the experiments of this paper the neural networks were trained with one of the training sets and then tested with the evaluation set in order to obtain the final results.

3. Proposed neural networks

All the models used in this paper to perform the test are neural networks based on one of three different architectures. Despite the different architecture, all the models share the following properties:

- They use Adam as their optimizer.
- Every unit that does not belong to the output layer uses ReLU as its activation function.
- The output layer is fully connected and has 527 units.

Since the audio event classification is a multi-labelled problem (i.e. the same segment can, and typically, contain more than one type of audio), all the networks (except one, as will be discussed later) use binary cross-entropy as their loss function and the binary sigmoid as the activation function for the output layer.

3.1. LSTM.

Since the data has the form of a time series of 10 feature vectors (embeddings), each one representing one second of audio, a Long Short-Term Memory (LSTM) recurrent neural network was immediately considered as an appropriate network, as they are specialized in this kind of data. The proposed LSTM model has the following architecture:

- The inputs are series of 10 128-dimensional feature vectors (embeddings).
- The first hidden layer is a unidirectional LSTM layer with 600 units, which outputs a single vector when the whole sequence has been processed. This layer is followed by a dropout layer with a 0.3 probability.
- The last hidden layer is a fully connected layer with 600 units.

3.2. CNN.

Convolutional Neural Networks (CNN) have proven to be very powerful for processing images. In our case, the input sequence of 10 128-dimensional feature vectors can be considered as a 10x128 image or matrix, and therefore a CNN could be appropriate in this case as well. The 128-dimensional input vectors are themselves produced as the output of a different neural network. This neural network uses a PCA transformation to create an embedding of the data. Therefore, there is no local proximity relationship between the elements of the vector. However, there is a temporal proximity for each individual feature, which translates into a local proximity between the rows of the matrix. A convolutional neural network with the following architecture was designed to take advantage of this temporal proximity:

- The input is the previously mentioned 10x128 matrix.
- The first hidden layer is a convolutional layer with 16 filters and a kernel with dimension 3x1. Since there is not proximity relationship in the input vectors (columns of the input matrix) the second dimension of the kernel is always restricted to 1 in our tests.
- The second hidden layer is a maximum pooling layer with a 2x2-dimensional window followed by a dropout layer with a 0.3 probability.
- The last hidden layer is a fully connected layer with 600 units.

3.3. MLP

Multi Layer Perceptrons (MLPs) are amongst the most standard and versatile neural networks. In fact, MLPs can approximate any input-output multidimensional output, including those produced by other network architectures, so they can be used as a reference model. The main advantage of other architectures over MLPs is that MLPs include a huge amount of weights, which can make training more difficult and more prone to overfitting. Given their property of

universal function approximation, they can be used to test the impact of other factors apart from the type of neural network they are based on. For this last reason, several models based on MLPs were developed and tested:

- Two MLPs with one hidden layer.
- A MLP with two hidden layers.
- A MLP with three hidden layers.

All these models share these properties:

- The input is a 1280-dimensional vector (the flattened version of the input matrix used for CNNs).
- The hidden layers have 1500 units each. After each one of them, there is a dropout layer with a 0.3 probability.

One of the MLPs with one hidden layer has the following particular properties:

- The hyperbolic tangent (tanh) is used as the activation function of the output layer.
- In this case, the Mean Squared Error (MSE) is used as the networks' loss function.

4. Test Description

In order to compare the performance of the different models, we evaluated them on the evaluation subset of Google Audio Set. Every model was trained with the balanced training test of Audio Set. In the case of the MLP with the bipolar sigmoid activation function, the target vectors were preprocessed so that they have a bipolar codification (the absence of a class is represented with the value -1 instead of 0), allowing us to test the effect of the codification of the data in performance.

All the models were trained with a minibatch size of 128. The training had a maximum duration of 50 epochs, however, early stopping was used in order to interrupt the training process when the mAP no longer increases for three epochs, thus preventing overfitting. No early stopping was used on the LSTM model as its mAP grew at a notably irregular rate and early stopping kept interrupting the training process before the model could reach its stability phase. This phenomenon didn't happen with the rest of the models.

As the main focus of these tests is to compare the different network architectures, hyper-parameters were left at their default values (learning rate: 0.001, beta1: 0.9, beta2: 0.999, decay: 0).

After training, the networks were tested with the evaluation subset of the Google Audio Set, and the final results were obtained by calculating the mean Average Precision (mAP) and the mean Recall (mR).

In addition, another test was performed on all the models based on MLPs where they were trained with the unbalanced training set (much larger in terms of samples, but much more unbalanced too) instead of the balanced one.

5. Results

After performing the tests described above, results presented in Table 1 were obtained:

Table 1: *Classification of the models ordered by their performance (in increasing mAP or mR order). 1h, 2h, 3h indicates the number of hidden layers. bal. unbal. Indicates the training set (balanced or unbalanced training set). bip. bin. Indicates the codification of the targets (bipolar or binary) and correspondingly the activation function of the output layer (tanh or sigmoid).*

Model, training set, codification	**mAP**	**mR**
MLP 1 h. l., bal., bip.	0.13704	0.15848
MLP 1 h. l., unbal., bip.	0.19696	0.22697
MLP 3 h. l., unbal., bin.	0.20686	0.23529
MLP 2 h. l., unbal., bin.	0.21203	0.24079
MLP 1 h. l., unbal., bin.	0.21342	0.24166
MLP 1 h. l., bal., bin.	0.21893	0.24249
CNN, bal., bin.	0.22830	0.25595
MLP 2 h. l., bal., bin.	0.24422	0.27542
MLP 3 h. l., bal., bin.	0.25276	0.28706
LSTM, bal., bin.	0.26652	0.30698

The first point to note is that the ranking of the different neural networks is the same whether the models are ordered by their mAP or their mR. Because of that, the metric considered is irrelevant when interpreting the results.

The models using bipolar data obtain the worst results. One possible reason could be that hidden layers use an activation function unable to take negative values. However, by comparing both models using bipolar data, we can notice that the one using the unbalanced training set has a much better performance than the one using the balanced training set. This is surprising because the models using binary data show the exact opposite behavior. In these models the use of the unbalanced training set has a negative impact on their performance, the effect becoming more intense the more layers the network has.

The MLPs using binary data and the balanced training set have a better performance the more hidden layers they have, which is the expected behavior when there is enough training data, as it seems to be the case.

The CNN's results were quite limited, falling behind the MLPs with more than one hidden layer. This is probably because of the lack of local meaning of the different features included in the 128-dimensional feature vectors (embeddings), which limits the kernel to a single dimension.

Finally, the model with the best results is the LSTM network, with a mAP of 0.26652 and a mR of 0.30698. This result is interesting because, even knowing that LSTM neural networks are particularly effective with time series, in our case these time series are very short, with 10 elements, which could have limited the performance of this model.

6. Conclusions and future work

After testing the performance of several deep neural networks, we were able to obtain a mAP of 0.26652 with a simple LSTM network. Despite these results being worse than the 0.314 mAP of the baseline established by Google, they allow us to draw some conclusions about creating models for Google Audio Set.

First of all, we can conclude that LSTM networks are the most appropriate architecture for this problem from all of those which were tested, as a relatively simple network with one LSTM layer and a fully connected layer offered better results than a more complex network with three fully connected layers, therefore recurrent neural networks should be a good starting point if a better performance is looked for, for example by adding more layers to the model or implementing more complex architectures.

The use of the balanced training subset seems to improve the performance of the models despite being less than 1/20 of the dataset. However, the unbalanced training subset was only used on MLPs due to time restrictions. Its effects on the other architectures should be studied in the future.

Transforming the target vectors to a bipolar codification decreases the performance of the models; however there seems to be a positive correlation between this codification and the use of the unbalanced training set, which could be worth researching into.

7. Acknowledgements

This work has been partially supported by project “DSSL” (TEC2015-68172-C2-1-P), funded by the Ministry of Economy and Competitivity of Spain and FEDER.

8. References

[1] KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems. 2012. p. 1097-1105.

[2] TEMKO, Andrey, et al. CLEAR evaluation of acoustic event detection and classification systems. In International Evaluation Workshop on Classification of Events, Activities and Relationships. Springer, Berlin, Heidelberg, 2006. p. 311-322.

[3] SALAMON, Justin; JACOBY, Christopher; BELLO, Juan Pablo. A dataset and taxonomy for urban sound research. In Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014. p. 1041-1044.

[4] IEEE AASP 2018 Challenge on Detection and Classification of acoustic Scenes and Events, http://dcase.community/challenge2018/index, (accessed: 05/10/2018).

[5] Freesound database, https://freesound.org, (accessed: 05/10/2018).

[6] CAKIR, Emre, et al. Polyphonic sound event detection using multi label deep neural networks. In *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015. p. 1-7.

[7] PARASCANDOLO, Giambattista; HUTTUNEN, Heikki; VIRTANEN, Tuomas. Recurrent neural networks for polyphonic sound event detection in real life recordings. *arXiv preprint arXiv:1604.00861*, 2016.

[8] Audio Set, a large scale dataset of manually annotated audio events, https://research.google.com/audioset/, (accessed: 20/05/2018).

[9] GEMMEKE, Jort F., et al. 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. p. 776-780.

[10] KONG, Qiuqiang, et al. Audio Set classification with attention model: A probabilistic perspective. *arXiv preprint arXiv:1711.00927*, 2017.

[11] YU, Changsong, et al. Multi-level Attention Model for Weakly Supervised Audio Classification. *arXiv preprint arXiv:1803.02353*, 2018.

[12] Audio Set Users Group, https://groups.google.com/forum/#!forum/audioset-users, (accessed: 20/05/2018).