CSCI 6908-Deep Learning Summer 2018

Project: Audio Tagging

Submission date: 07/Ago/2018

Student ID: B00735030 - Student Name: Miria Rafante Bernardino

Contents

Abstract

Introduction

The audio tagging problem can englobe different subproblems each with great variance of the complexity and difficulty each solution requires. Tagging environmental audios can be one of the most difficult once it must deal with noises and sounds that are not of interest and detect, above everything, when the sound of interest starts happening **Error! Reference source not found.**.

The problem focus of this work is a restricted and simplified problem. The sounds were recorded completely noiseless and were tagged into forty-one classes allowing the application of supervised machine learning techniques [1].

Literature review

Automatic tagging has been a desired solution as showed by the DCASE - Detection and Classification of Acoustic Scenes and Events [1]'s challenges, that have worked this problem for a few years already, each year with a subtle different focus [5][6] (http://www.cs.tut.fi/sgn/arg/dcase2017/) which have contributed to motive several works on the area.

Hamel et al [9]'s work, for instance, compared mel-spectrum and MFCC, two forms to represent the sounds that will be explained in the Describing the data section, on building models for classification. They conclude that mel-spectrum performs better than MFCC and the use of PCA also improves the results. After stablishing this first steps, they compared pooling functions over time, variating the size of the windows and combining some of these functions. The pooling functions were mean, variance, maximum, minimum, etc., and they observed the best outcome came from the combination of the four mentioned pooling functions, besides the performance were greater with windows lengths around two to four seconds. Their final-best result presented 0.876-AUC when tested over a dataset called MagnaTagATune and it was obtained by adding a hidden layer before the pooling step.

Choi et al [7] also used mel-spectrum as input. In this case, the mel-spectrum was input into their CNN with no fully connected layers but only three to seven convolutions and pooling layers are employed in order to have an output with dimensions 1x1, exactly, and increasing the number of filters at each layer. The output layer's activation function used was sigmoid. They reach a 0.894 on the AUC curve testing their approach on the same MagnaTagATune dataset.

Lee et al [8][7], accordingly, compared STFT, MFCC and MFCC spectrum and observed the best approach was the one using mel-spectrum. For their more outstanding approach, they used mel-spectrum to describe the data, with six convolutions by layer, each convolution followed by a max-pooling function. The mel-spectrum data were normalized by subtracting the average and dividing by its standard deviation. The filter length was fixed and had size of 243 values with stride of 81. They also used batch normalization and ReLU activation for the hidden layers and sigmoid

for the output layer. The cost function used was cross entropy. In the last convolution layers' output, they applied dropout of 0.5, stochastic gradient descent and 0.9 of momentum. The initial learning rate was 0.01 and decrease by 5 at each 3 epochs without improvement. The batch size was 23 for one database and 50 for the other. Their algorithm overcame prior state-of-the-arts methods and reach an AUC of 0.9059 for the same database used by Hamel et al [9] and Choi et al [7], MagnaTagATune dataset, overcoming these two.

On the contrary, the recent work of Xu et al [10], uses two CNN's combined with three RNN's. Besides, the descriptions choose are spectrogram, raw data and MFB (Mel filter banks feature) in opposition to the MFCC (Mel- frequency cepstrum coefficient) used till now; all three with IMD (interaural magnitude differences) that is a linear measure to incorporate spatial features introduced by this work. Before proceeding with the learning step, each sample is segmented using a slide windows of 32ms with a hop of 16ms and reshape into different dimensions according with the data description. Then, the CNN's have 128 filters, one size for each data description. The CNN is followed by 3 RNN's, that end up into 500 ReLU units connected with seven sigmoid output units, one for each class. The metric used for evaluation is the EER and their best approach performed an EEU of 0.102 on average among the seven classes.

Aiming to compare Lee et al [8] work, with an AUC of 0.9059, against Xu et al. [10] work with an EEU of 0.102, Figure 1 shows that ROC CURVE is the plot of true positive rate (TPR) in function of the false positive rate (FPR) and the results are measured in AUC (area under the curve) that represents how well the model correctly classified the positive class compared to the false examples also classified as positives [11]. The EEU is the error, the point where the false positive rate (FPR) is equal to the false negative rate (FNR).

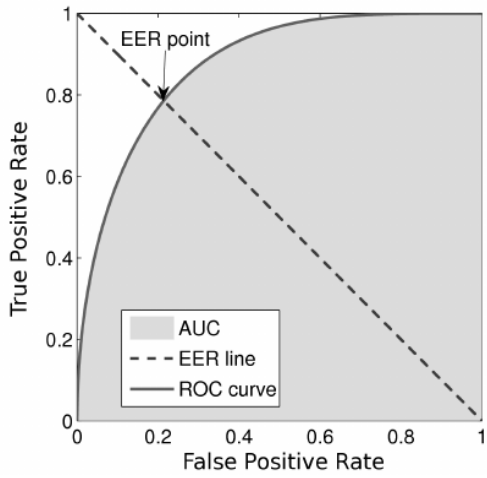

Figure 1 - Comparison of ROC curve, AUC and EER measurements. Image source: https://www.researchgate.net/figure/An-example-of-a-ROC-curve-its-AUC-and-its-EER_fig1_225180361

If we let the AUC be the percentage of correctly classified samples and EER be the percentage of misclassification samples, and ignoring that different datasets were used in each case, Lee et al

[8] and Xu et al. [10] works have similar results, with an error of 9.41% and 10.2%, with Lee [8] presenting an algorithm with a much simpler structure.

In this work, besides explore data descriptions and other classification algorithms, we also implement Lee [8] algorithm to compared with our own approaches.

Data Preparation

Dataset

The dataset is composed by 9473 samples unequally distributed among 41 categories. According to DCASE [1], "the minimum number of audio samples per category in the train set is 94, and the maximum 300. The duration of the audio samples ranges from 300ms to 30s ...". Considering the rate is 44.1 kHz, 44100 values per second, and that the recording time is different for each sample, the number of points per sample variety drastically. The greater sample has 1323000 values.

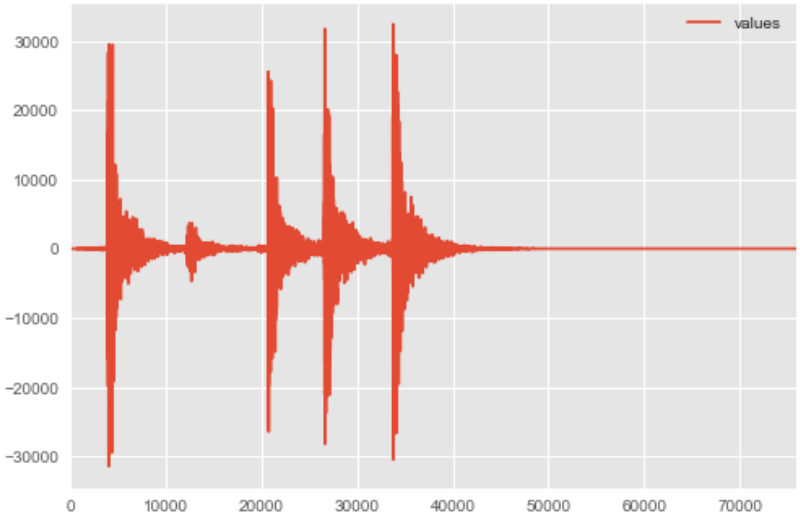

Figure 2 - Plotting a Knock sound sample. The x-axis is time in ms.

Figure 2 and Figure 3 show two samples' values (knock and a oboe) over time (ms).

Some of the samples' label were manually verified and some were not. For treat this problem we are using only the verified samples, in other words, we are using only the 3710 samples that were verified manually.

Sounds properties

Sounds are waves propagating over a physical matter [3]. The height (or amplitude) of the waves and the number of waves flowing by second (or frequency) are examples of the properties of a sound [2]. The higher the amplitude, the more energy the wave has, and intensity is the unity to measure the amount of energy a wave has in a given area [2]. Besides, tones, overtones, harmonics, speed of sound, timbre, loudness, etc., are other properties of the sounds and they variety according to its source [2][3].

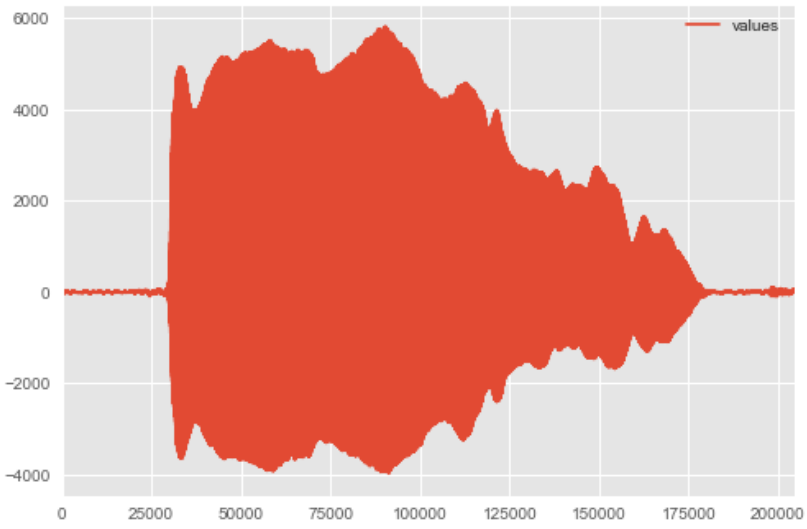

Figure 3 - Plotting a Oboe sound sample. The x-axis is time in ms.

Therefore, to identify the source of a sound, we need to study how the waves are changing over time and we also need a minimum period of time to be able to observe such changes [3], and we did that by describing the samples in function of some of its properties.

Describing the data

We used the 4 functions shown in Figure 4 and Figure 5 from librosa [4] package for feature extraction to describe the original data: tonnetz (computes the tonal centroid features (tonnetz)), spectral_centroid (computes a 6D description of chords), spectral_bandwidth and mfcc (computes the Mel-frequency cepstral coefficients (MFCCs)).

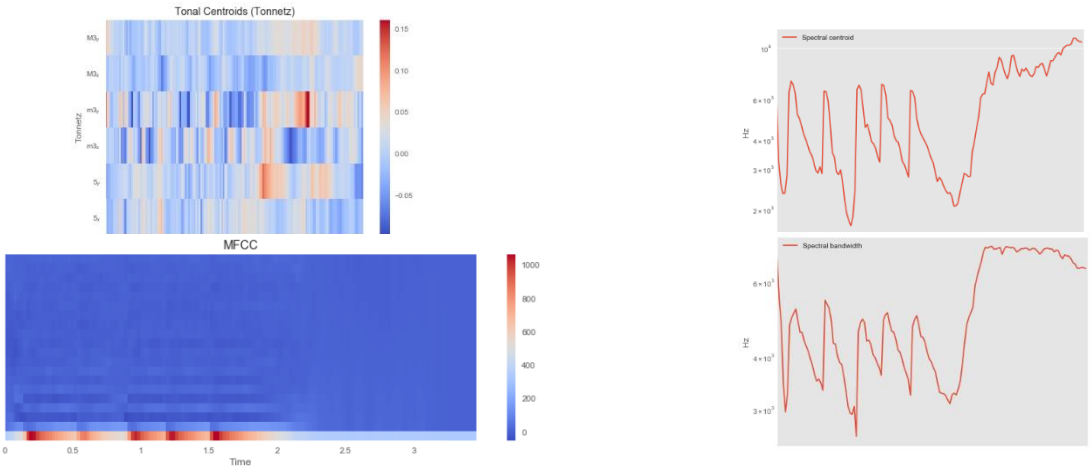

Figure 4 - Vizualization of the knock sample trough librosa's functions for feature extraction: (top-left) Tonnetz, (top-right) Spectral centroid, (bottom-left) MFCC and (bottom right) Spectral bandwidth. Source: https://librosa.github.io/librosa/index.html

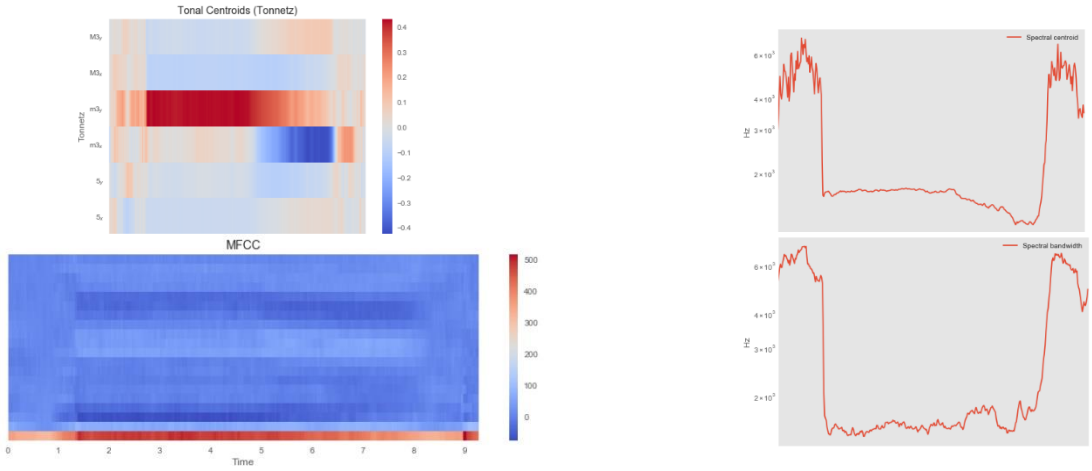

Figure 5 - Visualization of the oboe sample trough some librosa's functions for feature extraction: (top-left) Tonnetz, (top-right) Spectral centroid, (bottom-left) MFCC and (bottom right) Spectral bandwidth. Source: https://librosa.github.io/librosa/index.html

####Explain a little bit a about each function and mention the format of the output and what do they mean###

The fifth description we tried was the spectrogram. The function is the stft function from the librosa [4] package and creates a 3-dimensional space combining the time series in milliseconds (ms) X 1025 frequencies X the intensity in decibels (dB) at each time-frequency

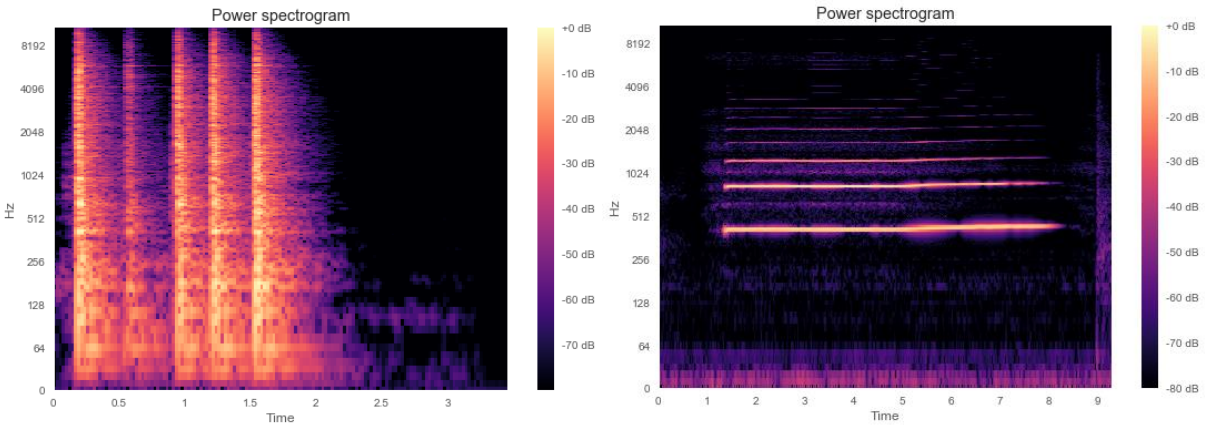

Figure 6 - Power spectrograms of a Knock and an Oboe sound samples, respectively. Dimensions: 1025 frequencies x time x intensity at each time-frequency point.

Finally, considering Lee et al. [8], we also describe the data into mel-spectrograms. The plots of two sounds samples are shown in Figure 8.

The mel-spectrogram calculates the spectrograms on the mel-frequency. The output dimensions for each sample are 128 frequencies X time in milliseconds (ms) X the intensity in decibels (dB) at each time-frequency. This spectrogram is much smaller than the former one, presenting only 128 frequencies against 1025 and that was another reason to input this description into the classification approach.

In order to have a first result over this new representation of the data, we proceed analysing the size of the samples which is proportional to its length in ms (Figure 7), and decided to limit the size to 1000 for all samples, once 90% of the samples in use are smaller than that, padding with zeros the samples that are smaller than that and reshaping it to one dimension.

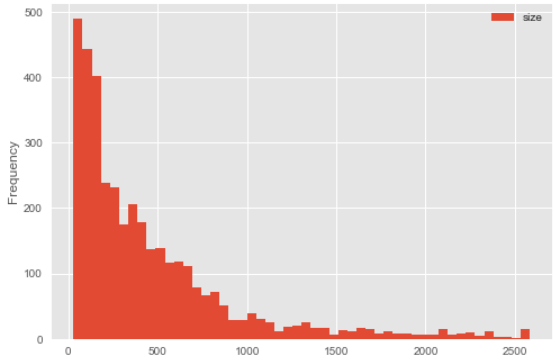

Figure 7 - Number of samples of each size (size is proportional to the samples' length in ms).

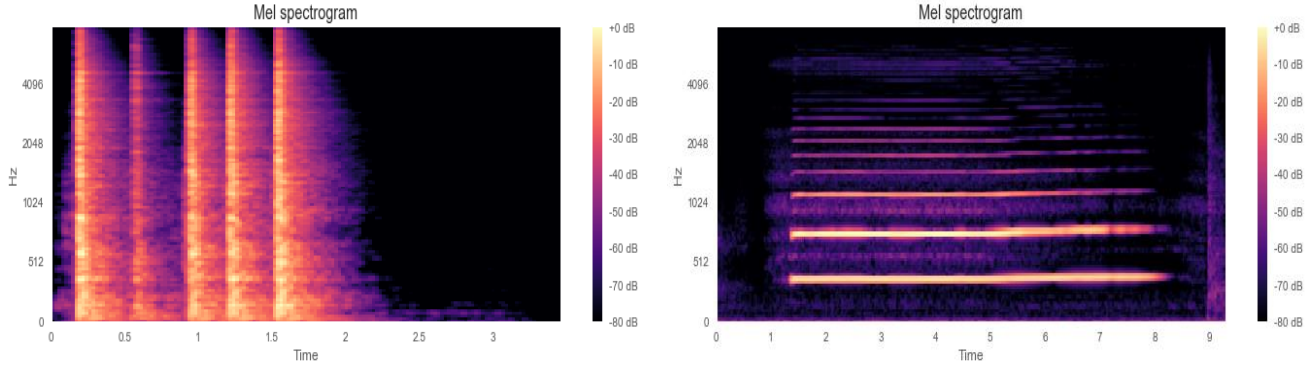

Figure 8- Mel spectrograms of a Knock and an Oboe sound samples, respectively. Dimensions: 128 frequencies x time x intensity at each time-frequency point.

Methods

2 methods so far:

from sklearn.neural_network import MLPClassifier

clfMLP = MLPClassifier(

```
    hidden_layer_sizes = (2560,1024,256),
    activation = 'relu',
    solver = 'adam',
    batch_size = 50,
    learning_rate = 'invscaling',
    learning_rate_init =0.3,
    tol = 1e-3,
    verbose = True,
    )
from sklearn.ensemble import RandomForestClassifier
clfRF = RandomForestClassifier(n_estimators = 1000, criterion="entropy", n_jobs=-1)
```

Results

Data description	MaxSize	RF – 3-fold	MLP- 3-fold
Tone	1000	Scores: [0.2971246, 0.31012146, 0.2968 1112] Mean: 0.30135239277503162 Std: 0.006201985675404002	Scores: [0.0686901, 0.06720648, 0.069 50123] Mean: 0.068465933357223716 Std: 0.00095014194851230651
Spectral Centroid	1000	Scores: [0.43210863, 0.43724696, 0.448 0785] Mean: 0.4391446950878993 Std: 0.006656336335106039	Scores: [0.0686901, 0.06882591, 0.06 78659] Mean: 0.06846063676458795 Std: 0.00042417933477714254
Spectral Bandwidth	1000	Scores: [0.44888179, 0.43076923, 0.43 254293] Mean: 0.4373979823782462 Std: 0.008152499127789991	Scores: [0.0686901, 0.06720648, 0.06 950123] Mean: 0.06846593335722372 Std: 0.00095014194851230651
MFCC	1000	Scores: [0.64616613, 0.65101215, 0.654 94685] Mean: 0.65070837731252074 Std: 0.0035911426034004267	-

Data description	MaxSize	RF – 10-fold
MFCC	1000	Scores: [0.67792208, 0.66318538, 0.67810026, 0.63852243, 0.7037037, 0.67204301, 0.67847411, 0.66111111, 0.64145658, 0.69714286] Mean: 0. 67116615275898628 Std: 0. 019986117622205764
Mel-spectrogram	1000	

Conclusion

References

[1] Dcase. "General-purpose Audio Tagging of Freesound Content with AudioSet Labels." General-purpose Audio Tagging of Freesound Content with AudioSet Labels - DCASE. Accessed July 19, 2018. http://dcase.community/challenge2018/task-general-purpose-audio-tagging.

[2] "The Components of Sound." Conductors and Insulators. Accessed July 19, 2018. https://www.nde-ed.org/EducationResources/HighSchool/Sound/components.htm.

[3] "Sound." Wikipedia. July 15, 2018. Accessed July 19, 2018. https://en.wikipedia.org/wiki/Sound.

[4] "LibROSA." LibROSA - Librosa 0.6.0 Documentation. Accessed July 19, 2018. https://librosa.github.io/librosa/index.html.

[5] S. Yun, S. Kim, S. Moon, J. Cho, and T. Kim, “Discriminative training of gmm parameters for audio scene classification and audio tagging,” IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2016).

[6] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, “CP-JKU submissions for dcase-2016: A hybrid approach using binaural i-vectors and deep convolutional neural networks,” Tech. Rep., DCASE2016 Challenge, Tech. Rep., 2016.

[7] Choi, K., Fazekas, G., & Sandler, M. (n.d.). AUTOMATIC TAGGING USING DEEP CONVOLUTIONAL NEURAL NETWORKS. Retrieved from https://arxiv.org/pdf/1606.00298.pdf

[8] Lee Jiyoung Park Keunhyoung Luke Kim Juhan Nam, J. (n.d.). SAMPLE-LEVEL DEEP CONVOLUTIONAL NEURAL NETWORKS FOR MUSIC AUTO-TAGGING USING RAW WAVEFORMS. Retrieved from https://github.com/keunwoochoi/MSD_split_for_

[9] Hamel, P., Lemieux, S., Bengio, Y., & Eck, D. (n.d.). TEMPORAL POOLING AND MULTISCALE LEARNING FOR AUTOMATIC ANNOTATION AND RANKING OF MUSIC AUDIO. Retrieved from

http://www.iro.umontreal.ca/~lisa/bib/pub_subject/finance/pointeurs/music_pooling.pdf

[10] Xu, Y., Kong, Q., Huang, Q., Wang, W., & Plumbley, M. D. (2017). Convolutional gated recurrent neural network incorporating spatial features for audio tagging. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 3461–3466). IEEE. https://doi.org/10.1109/IJCNN.2017.7966291

[11] Schoonjans, Frank. ROC Curve Analysis with MedCalc. MedCalc. May 09, 2017. Accessed July 26, 2018. https://www.medcalc.org/manual/roc-curves.php.