Audio Processing and Indexing

Bird's sound classification

Georgios Kyziridis
s2077981
g.kyziridis@umail.leidenuniv.nl

January 20, 2018

Abstract

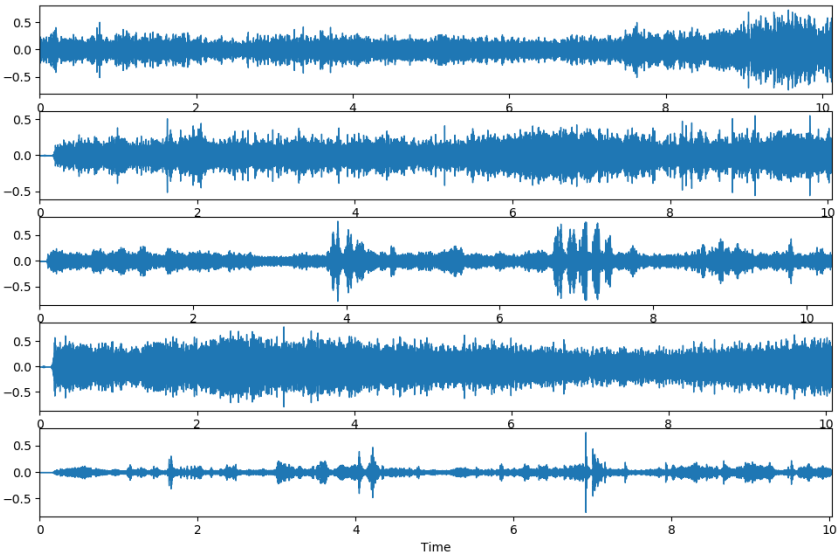
Nowadays urban sound classification approaches are becoming more and more state of the art in the scientific field. There is a lot of scientific research on the specific subject of how the sound is suppose to be classified and detected by machines. This fact gives the opportunity to the users to use interesting applications which recognize and detect sounds such as Shazam. That kind of applications that detecting real sounds make life easier as they answer simple questions for example "what song is this?" or "what kind of animal is this" according to the raw sound. This paper addresses some challenging approaches for bird detection according to their twit sounds.

1 Introduction

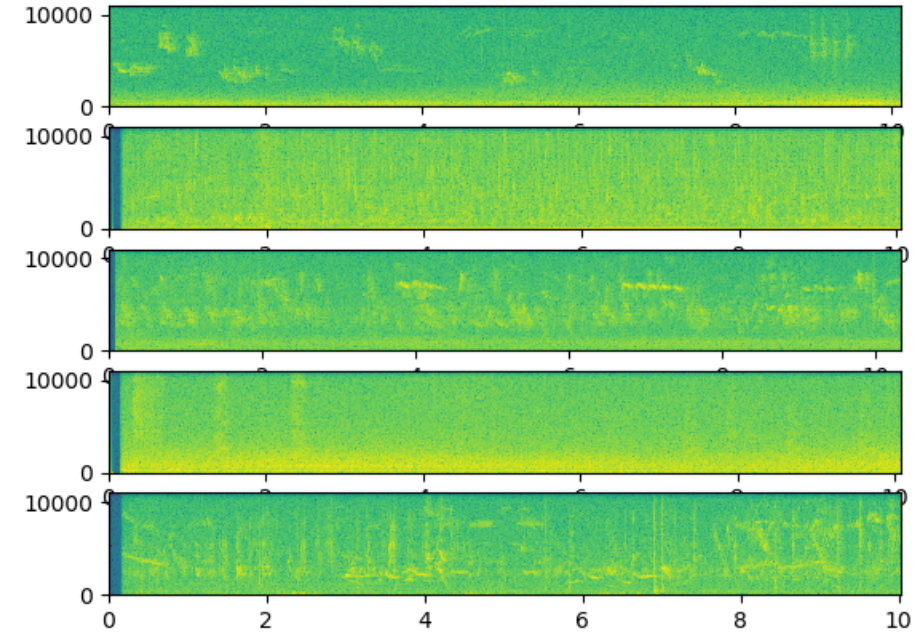
The following sheet is an endeavor on trying to explain the procedure of binary classification problem in a big repository of wav files with bird sounds. The goal of this project is to construct a robust detection/classification algorithm in order to make the machine to learn how to classify if the sound (wav) corresponds to a real bird or not.

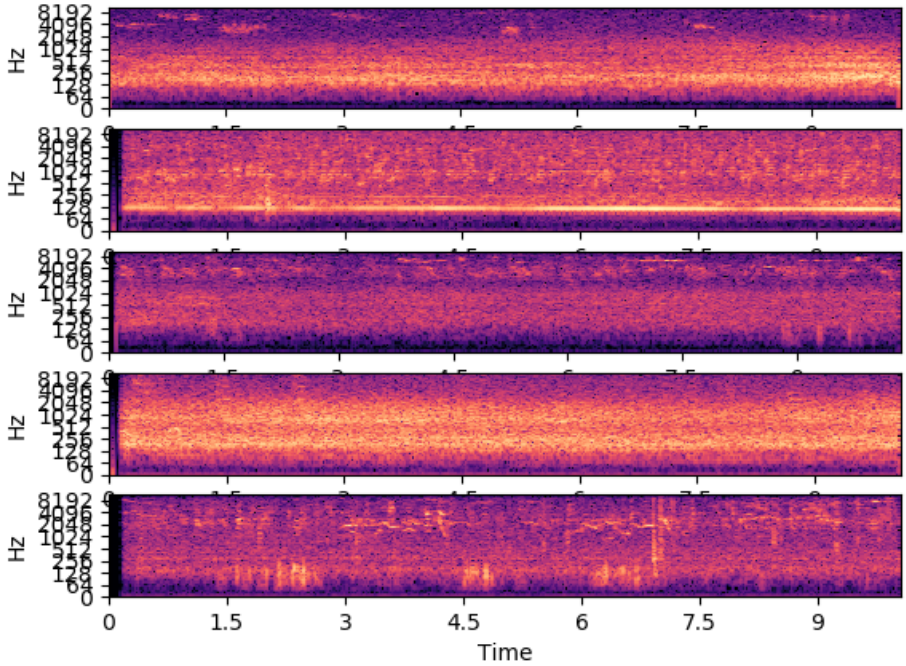
2 Data Exploration

The specific dataset includes 8000 wav of bird-sounds. Some of the waves are shown below.



The graphs below provide spectograms and logspectograms for the previous five waves.





We can easily define the dissimilarity between the raw sounds and the fact that the sounds are distorted which will be an obstacle on producing robust predictions.

3 Methods

There are many approaches and methods used for sound classification. This paper is based on Neural Networks implementation for binary classification fed in with various sound features with respect on fast fourier transformation. The two methods which were implemented in this project are simple neural network and recurrent neural network. Both of them are part of Neural Networks using Fast Fourier Transformation of the raw sound as input.

3.1 Simple Neural Network

The first method which used in this classification problem was a simple Neural Network with four layers. The network fed in with a concatenation of many sound-features such as:

- **melspectrogram**: Mel-scaled power spectrogram
- **mfcc**: Mel-frequency cepstral coefficients
- **chorma-stft**: A chromagram from a waveform or power spectrogram
- **spectral_contrast**: Spectral contrast
- **tonnetz**: The tonal centroid features

Method in steps:

1. Split the initial repository of wavs into four smaller(56 , 150 , 500 , 8000)
2. Import the data
3. Extract features for each wav and concatenate them into a vector
4. Split dataset into train and test (70% - 30%)
5. Train NN for each smaller dataset and compare the results

Neural Network parameters

After a lot of searching and tunning differently the parameters and the functions of the neural network I used the cross-entropy cost function with the gradient descend optimizer which was the most efficient for binary classification. Moreover, after different tests in the same NN architecture I found that the efficient number of layers is three or four. I test it until the number of layers and neurons will not have an impact on the results. The final results after many different experiments in the same model are shown in the table.

Results

Num of wavs	Accuracy
56	76%
150	66%
500	19%

Table 1: 4-layer NN with 1500 iterations and 0.01 learning rate

Conclusion

The evaluation and the training for that naiv approach was done in a simple Lenovo laptop with an Intel(R) Core(TM) i5-3320M CPU @ 2.60GHz and 8GB RAM in GNU/Linux environment (Debian-8 Jessie) using python. We can observe in Table 1 that as the number of wavs is increasing the accuracy is decreasing. That fact drive us to the conclusion that the specific model is not predicting correctly with big amount of wavs. Probably this is reasonable because the method is naiv and uses as input the hole extracted information from the sound without considering the dissimilarity of the raw sound-data.

3.2 Recurrent Neural Network

The second approach was to implement a recurrent neural network. The idea behind it is that now the raw sound will be windowed and the feature of mel-frequency cepstral coefficient would be extracted from each window. So the input of the RNN will be time-series with a vector of mfcc for each timestamp(window). The significant element of Recurrent Neural Networks is that they use sequential information from the input data trying to predict the next event for each step. The most common approach of RNN is the LSTM which are gaining power from the long short-term memory of the prior information in order to produce a robust conditional prediction for the immediate future. RNN performs the same task for every element in the sequence with the output being depend on the prior information. Furthermore, that type of networks are based on the assumption that the input frames are all strong depended and the next event is based on the prior, that is why all layers and neurons are strongly bidirectional connected.

Experiment in steps:

1. Split the initial repository of wavs into four smaller(56 , 150 , 500 , 8000)
2. Import the data
3. Extract mfcc feature for each window in each wav
4. Split dataset into train and test (70% - 30%)
5. Train RNN for each smaller dataset and compare the results
6. Train the same model with the big initial repository(8000 wav) on "Duranium" super computer

RNN - parameters
In this approach of LSTM-RNN parameters were tuned as follows:

- learning rate = 0.01
- training iterations = 1000
- input size = 20
- number of steps = 41

Results

Num of wavs	Accuracy
56	24%
150	59%
500	60%
8000	77%

Table 2: RNN with 1000 iterations and 0.01 learning rate

Conclusion

As we can observe in Table 2 that the more input wavs the better accuracy as output. We can easily define that the recurrent neural network is working better than the standard feed-forward approach by understanding the complexity of the data structure. The fact that RNN are making robust estimations on time-series fit for sound-classification case because the feature extraction was done by windowing the raw sound and build time-series for the network input. Finally, the network provides output based on the conditional probability of $mfcc_{new}$ given the previous mfcc event (window).

4 General Conclusion

We can definitely conclude that LSTM-RNN approach was better than the standard naiv approach of feed-forward neural network. The basic fact that upgrade the results was that RNN architecture is based on the assumption that our data can be described in a sequential way as time-series and that each event on the time line is correlated with the previous one. Furthermore, using windowing and mfcc as the main feature extracted from the raw sounds, had a strong impact on the efficient classification-result of RNN.

5 Future Work

There are many different implementations of neural networks specified for sound detection/classification. A very interesting and efficient method is to construct the spectogram for each wav_sound and train a Convolutional Neural Network(CNN) for image classification. Nevertheless, another interesting idea based on the previous one is to feed the CNN with the log scaled mel-spectrograms sound clips and train the network like experimenting on image classification but instead of images with sound_clips. That could probably produced better estimations for the sound classification problem.

6 References

- Urban-Sound-Classification(github)
- Mel-frequency-cepstral-coefficients
- Audio content analysis by Juan Pablo Bello
- Bird audio detection challenge