Laboratory Report - Supervised audio classification

Course: SGN-26006 Advanced Signal Processing

Assignment no. 6: Image Recognition

Authors: Adam Ligocki, Amir Salah

Released: 17th of December, 2018

Table of Contents

[1. Task 3](#_Toc532813045)

[2. Modern solutions for audio classification 3](#_Toc532813046)

[3. Dataset 4](#_Toc532813047)

[4. Model 6](#_Toc532813048)

[Architecture 6](#_Toc532813049)

[Preparing data for training 7](#_Toc532813050)

[Learning 7](#_Toc532813051)

[5. Results 8](#_Toc532813052)

[6. Conclusion 9](#_Toc532813053)

[7. References 10](#_Toc532813054)

# Task

Investigate the techniques behind audio classification and build a neural network model that performs a supervised classification.

For this lab we choose to go for the subdomain of audio classification of the environment sounds. That’s why we picked the dataset UrbanSound8K. It offers sound segments sliced to at 4 seconds maximum.

Implementation was done in Python using Keras library that abstracts on top of TensorFlow. Conda virtual environment was used to run jupyter-notebook that will help to visualize each step and what was achieved.

# Modern solutions for audio classification

There are different approaches to audio classification nowadays. The choice usually comes down to what features you want to pick up from the signal, as the audio signal is rich with features that can be extracted.

If we are dealing with sounds that is targeted for humans, the range is limited from 0db, which is the minimum a human ear can hear, to around 120db at which point even a short exposure to a nearby audio source can cause severe permanent damage. When it comes to sound frequencies, the human ear is able to pick from about 15 to about 18,000 waves, or cycles, per second.

The audible sounds include speech, music, natural sounds, artificial sounds and noise. Each can be represented either in time domain or in frequency domain depending on the task at hand, and the filter to be used.

The audio features include physical features and perceptual features, the latter is what humans can easily distinguish such as the loudness and the pitch of the sound, however, these features are not commonly used in classification as they don’t contain relevant information about the signal, and can often be interchangeable between signals that have nothing in common.

The physical features however are low level features that forms the backbone of digital audio processing. This include Zero Crossing Rate, Short-Time Energy, Spectral Centroid, Spectral Roll-off, Spectral Flux, Fundamental Frequency and Mel-Frequency Cepstral Coefficient (MFCC).

The most used physical feature in speech and audio analysis is the former, Mel-Frequency Cepstrum. The mel-frequncy is a better representation of how the human auditory systems distinguish different frequencies. It seeks to resemble the response more closely than the linearly-spaced frequency bands used in the normal cepstrum.

Finally, when it comes to the actual classification algorithm, one of the common ways how this task is done is using a convolutional neural network. The same NN that was proven very successful for image recognition was shown to be very suitable for processing audio too. This can be done since an audio signal can be represented as a spectrogram which is a grid-like data similar to a digital representation of a picture.

Convolutional neural network can capture spectro-temporal patterns across time and frequency when fed with spectrogram data. In addition to that, they can also discriminate the sound even when it is masked by other noise, which is something that traditional Mel-Frequency Cepstral Coefficient suffers at. Therefore, in this task CNN were chosen for audio classification.

# Dataset

Urban8k dataset contains 8732 labeled sound excerpts, each of them is less than 4 seconds long. All sounds are from an urban environment and in total there are ten classes, labeled from 0 to 9:

0. air\_conditioner

1. car\_horn

2. children\_playing

3. dog\_bark

4. drilling

5. engine\_idling

6. gun\_shot

7. jackhammer

8. siren

9. street\_music

The files were divided into ten folders named from one to 10.

Together with the audio signals, the dataset include a CSV file containing metadata about each recording.

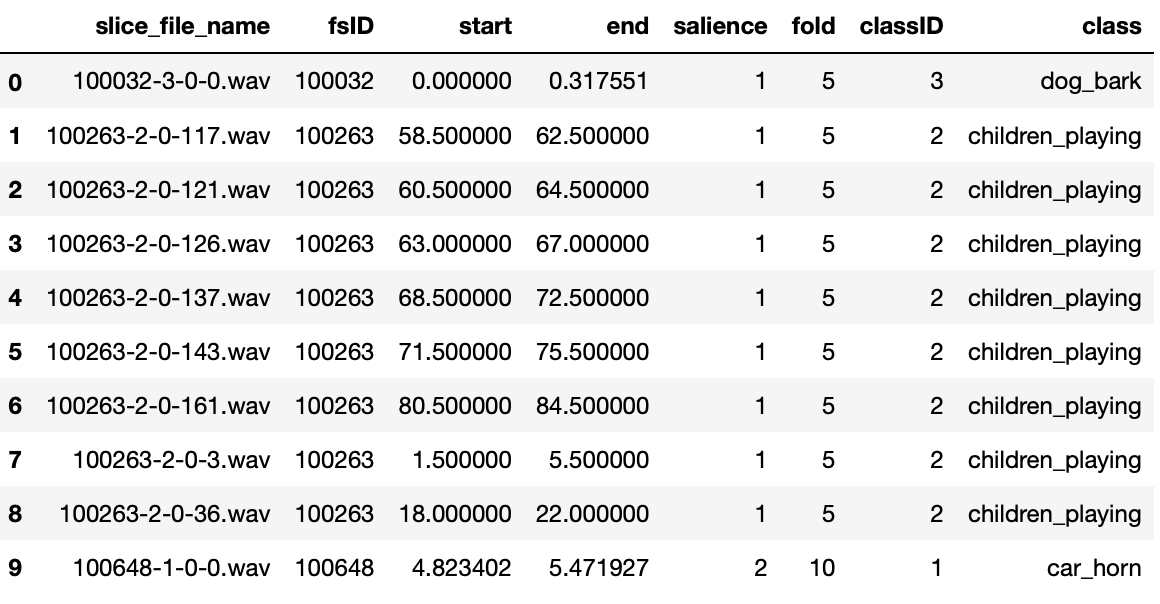


Figure 1 – First 10 rows of the dataset

According to the dataset website, this is what each column represents:

* slice\_file\_name: The filename taking the format format: [fsID]-[classID]-[occurrenceID]-[sliceID].wav, where:
  + [fsID] = the Freesound ID
  + [classID] = a numeric identifier of the sound class
  + [occurrenceID] = a numeric identifier to distinguish different occurrences of the sound within the original recording
  + [sliceID] = a numeric identifier to distinguish different slices taken from the same occurrence
* fsID: The Freesound ID of the recording from which this excerpt (slice) is taken
* start: The start time of the slice in the original Freesound recording
* end: The end time of slice in the original Freesound recording
* salience: A (subjective) salience rating of the sound. 1 = foreground, 2 = background.
* fold: The fold number (1-10) to which this file has been allocated.
* classID: A numeric identifier of the sound class that was already listed above.
* class: The class name

# Model

## Architecture

We have designed a convolutional multilayer neural network. It contains six 3 2Dconvolutional layers, in each layer Relu activation function was used. Every second layer is maxpooled with variable size shown below.

The final layer is fully connected one with softmax output into one-hot-end encoding ten classes output. These parameters follows the result of a research paper by Justin Salamon and Juan Pablo Bello listed in the references.

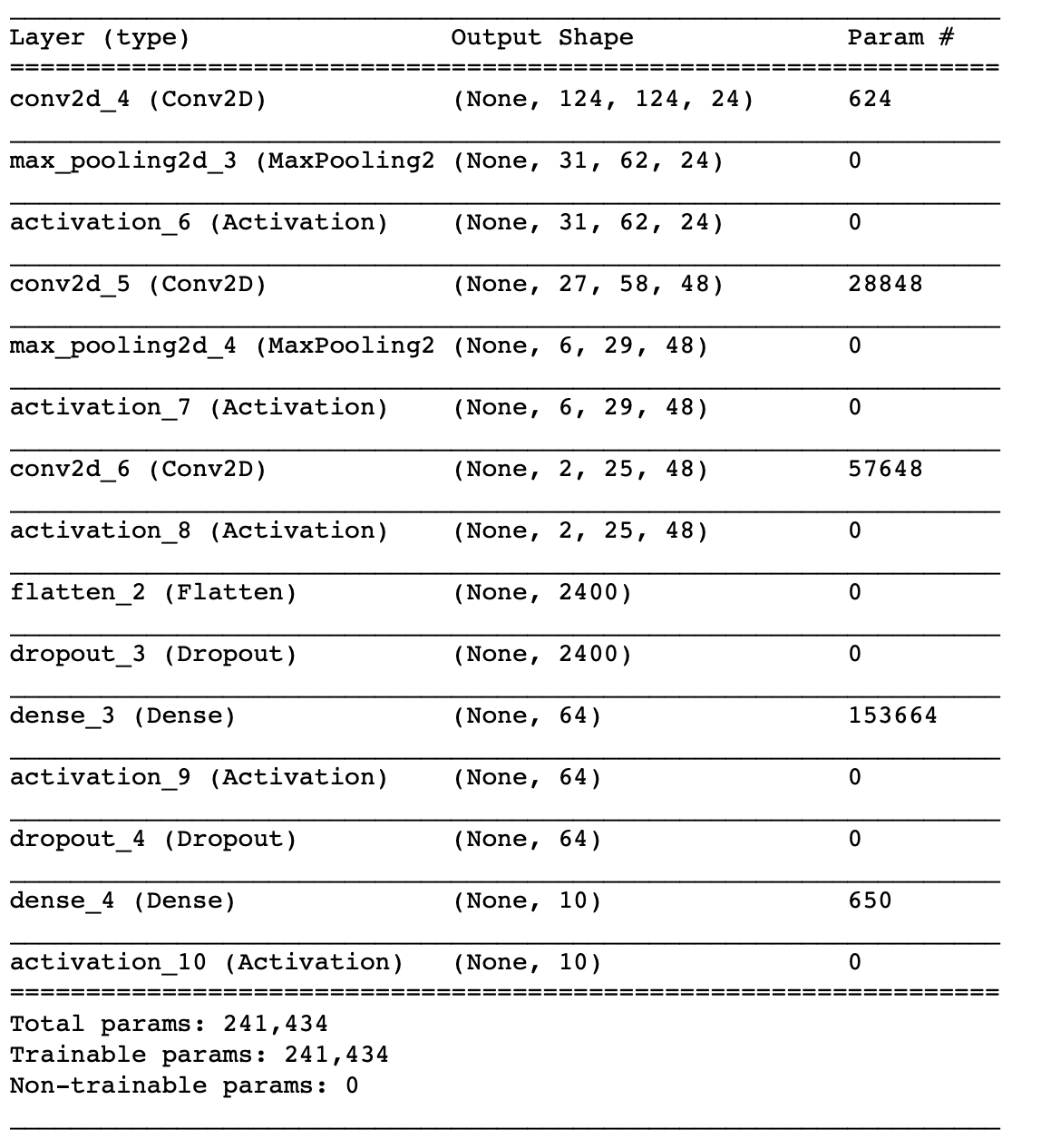
****

Figure 2 – Summary of the convolutional neural network model used

## Preparing data for training

In order to visualize our audio signal, we have to convert them to spectrograms of frequencies varying over time, specifically in this project we convert them into log scale mel spectrogram. This grid data can be processed by the convolutional neural network. Jupyter Notebook visualizes this process. The grid has 128 frequencies blocks and 128 timestamps. The input is then a 128x128 array of real numbers.

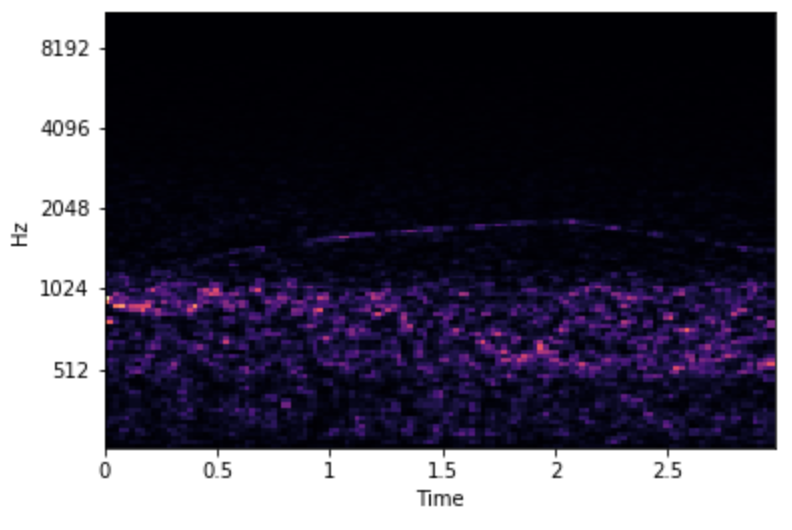


Figure 3 – Example of log scale mel spectrogram of Siren sound

## Learning

The learning process has been realized by Keras built-in fit function for sequential model.

For scoring the classification problem we have used the categorical crossentropy loss function and learning has been driven by Adam optimizer with variable initial parameters that we experimented with. Final ones are in the script.

The learning parameters has been chosen to be 500 batch size and 12 epochs.

There was no terminating condition set since the learning rarely reached 90% accuracy or higher.

# Results

First testing has been done on around 650 test signals. Every audio signal has been fed as input to the convolutional neural network and the output has been compared with the validation labels. As a result, we got the accuracy of 79.5%

We also tried the network with data from real life, for some audio signal we manage to recorded it from internet videos using smartphones, for some other effects we directly recorded using phone outside. The filename of the audio file was chosen in a way that the first digit of the name represents the class number that the audio file should belong to.

The results came out positive in majority of cases with the exception of gunshots sounds effects and jackhammer sound. After investigating, it appears that the reason the gunshots sound was not detected was because all the audio samples were recorded for a single shot only while the sound we recorded from a YouTube video originated from a spree of firing. Notice below the difference in the spectrogram which makes it impossible for the CNN to classify as the same.

|  |  |
| --- | --- |
|  |  |

Figure 4 – On the left, spectrogram of single firing, on the right, spectrogram of multiple gunshots.

As for the other error, it was the CNN confusing the jackhammer sound for the drilling sound. To be fair, this one is even hard for a human to distinguish when you listen to both sounds, and their spectrograms have slightly similar patterns.

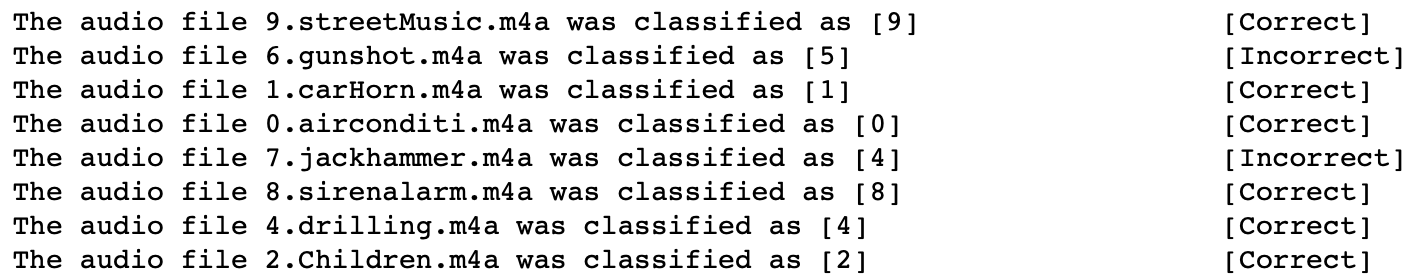


Figure 5 – Results while experimenting with real life data.

# Conclusion

In this assignment, we have successfully designed and trained a convolutional neural network for classifying audio signals. Our network consists of 3 convolutional layers combined with max pooling layers. The output is finalized with two fully connected layers. Output is in ten-hot-encoding format.

We have trained this network using 6800 labeled signals, validation was done on 650 signals.

The test results give us an accuracy of around 79.5%.

At the end, the network was also tested with real data and it came out accurate on most of the input signals with some exception.

There are more advanced techniques to improve the result of our trained model, one of them, and perhaps the most guaranteed way is to have a training over a larger dataset. Our dataset of around 8000 sound signals is relatively small. However, the hardware limitations are quick to reach when dealing with such a task. The laptop this task was done often took around an hour to only append the data to a list placeholder. However, if these limits were not a problem, one way to enhance the accuracy even with the lack of new data is to use a technique called data augmentation.

As the name reveals, data augmentation is the process of using the same dataset but with tweaking some features by introducing some distortion to the old signals to generate new ones. As long as the amount of distortion is not too high so the data remains valid for the classification.

Different techniques exist to augment data, for audio, some of the popular ones include adding gaussian noise with a low random distribution, cropping different part of the audio signal, varying the speed of the audio so that it’s either faster or slower, and thus has a different length. Some advanced techniques also include the use of GAN networks to generate new data.

# References

[1] Audio Signal Processing: A Review of Audio Signal Classification Features. Available from: https://www.researchgate.net/publication/318224324\_Audio\_Signal\_Processing\_A\_Review\_of\_Audio\_Signal\_Classification\_Features [accessed Dec 12 2018].

[2] Venkatesh Boddapatia, Andrej Petefb, Jim Rasmussonb, Lars Lundberg. Classifying environmental sounds using image recognition networks. https://ac.els-cdn.com/S1877050917316599/1-s2.0-S1877050917316599-main.pdf?\_tid=1d08dc8f-9c15-42e7-b93a-5bfd7ff4ac65&acdnat=1545011770\_33a55580c8d9a36a0632637761e5c5de

[3] Justin Salamon, Christopher Jacoby, Juan Pablo Bello. A Dataset and Taxonomy for Urban Sound Research http://www.justinsalamon.com/uploads/4/3/9/4/4394963/salamon\_urbansound\_acmmm14.pdf

[4] URBANSOUND8K dataset. https://urbansounddataset.weebly.com/urbansound8k.html. [accessed Dec 11 2018].

[5] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore. CNN ARCHITECTURES FOR LARGE-SCALE AUDIO CLASSIFICATION. https://arxiv.org/pdf/1609.09430.pdf [accessed Dec 13 2018].