Udacity Machine Learning Nanodegree

Capstone Proposal – Determine Different Sounds using Deep Learning

By

Ayush Saxena

12th August, 2019

➔ Domain Background

In our day to day life, we always use our all 5 sense on every milliseconds to judge what is happening in our surrounding. The one most important sense it Hearing which help use listening the voice or sounds and predict the state what it wants to tell us so we can take action accordingly. Sounds are all around us. Whether directly or indirectly, we are always in contact with audio data. Sounds in our daily life can be, conversation with someone, music, Video with audio, other environmental sounds like car, wind, construction, machines working in factories and industries, background noise. The human brain is continuously processing and understanding this audio data, either consciously or subconsciously, giving us information about the environment around us.

Automatic environmental sound classification is a growing area of research with numerous real world applications. Whilst there is a large body of research in related audio fields such as speech and music, work on the classification of environmental sounds is comparatively scarce. Likewise, observing the recent advancements in the field of image classification where convolutional neural networks are used to classify images with high accuracy and at scale, it begs the question of the applicability of these techniques in other domains, such as sound classification, where discrete sounds happen over time.

The goal of this capstone project, is to apply Deep Learning techniques to the classification of environmental sounds, specifically focusing on the identification of particular urban sounds.

Some most common real world applications, such as: - Content-based multimedia indexing and retrieval - Assisting deaf individuals in their daily activities - Smart home use cases such as 360-degree safety and security capabilities - Automotive where recognising sounds both inside and outside of the car can improve safety - Industrial uses such as predictive maintenance

I am working in the field of analytics and music is one of my passion. I love to play guitar and flute. Most of the time when I travel I used to identify the sounds in music of songs or the sounds of my surroundings'. I am keen to apply my machine learning knowledge to this domain as it would help me in creating an application for sound detection.

➔ Problem Statement

The main objective of this project will be to use Deep Learning techniques to classify urban sounds. When given an audio sample in a computer readable format (such as a .wav file) of a few seconds duration, we want to be able to determine if it contains one of the target urban sounds with a corresponding likelihood score. Conversely, if none of the target sounds were detected, we will be presented with an unknown score.

➔ Datasets and Inputs

For this project we will use a dataset called Urbansound8K [1]. The dataset contains 8732 sound excerpts (<=4s) of urban sounds from 10 classes, like: Air Conditioner, Car Horn, Children Playing, Dog bark, Drilling, Engine Idling, Gun Shot, Jackhammer, Siren Street Music. The classes are drawn from the urban sound taxonomy.

In addition to the sound excerpts, we will have a CSV file (UrbanSound8k.csv) containing metadata about each excerpt is also provided.

This file contains meta-data information about every audio file in the dataset. This includes:

slice_file_name:
The name of the audio file. The name takes the following format: [fsID]-[classID]-[occurrenceID]-[sliceID].wav, where:
[fsID] = the Freesound ID of the recording from which this excerpt (slice) is taken
[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

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:
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

Class:
The class name: air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gun_shot, jackhammer,
siren, street_music.

The accompanying metadata contains a unique ID for each sound excerpt along with it's given class name. These sound excerpts are digital audio files in .wav format. Sound waves are digitised by sampling them at discrete intervals known as the sampling rate (typically 44.1kHz for CD quality audio meaning samples are taken 44,100 times per second). Each sample is the amplitude of the wave at a particular time interval, where the bit depth determines how detailed the sample will be also known as the dynamic range of the signal (typically 16bit which means a sample can range from 65,536 amplitude values). Therefore, the data we will be analysing for each sound excerpts is essentially a one dimensional array or vector of amplitude values.

➔ Solution Statement

The proposed solution to this problem is to apply Deep Learning techniques that have proved to be highly successful in the field of image classification.

First we will extract Mel-Frequency Cepstral Coefficients (MFCC) [2] from the audio samples on a per-frame basis with a window size of a few milliseconds. The MFCC summarises the frequency distribution across the window size, so it is possible to analyse both the frequency and time characteristics of the sound. These audio representations will allow us to identify features for classification.

The next step will be to train a Deep Neural Network with these data sets and make predictions. This will be very effective at finding patterns within the MFCC's much like they are effective at finding patterns within images.

We will use the evaluation metrics described in later sections to compare the performance of these solutions against the benchmark models in the next section.

➔ Benchmark Model

For the benchmark model, we will use the algorithms outlined in the paper "A Dataset and Taxonomy for Urban Sound Research" (Salamon, 2014) [3]. The paper describes five different algorithms with the following accuracies for a audio slice maximum duration of 4 seconds.

Algorithm	Accuracy
SVM_rbf	68%
RandomForest500	66%
IBk5	55%
J48	48%
ZeroR	10%

➔ Evaluation Metrics

The evaluation metric for this problem is simply the Accuracy Score.

➔ Project Design

Data Pre-processing:

First identify the different data types in our dataset and what pre-processing needs to be done to make it uniform.

- Re-sample so all audio had the same sample rate and bit depth
- Make sure the sample duration is uniform

- Consider any data augmentations, such as adding background noise (though this maybe a nice to have)

Data Splitting:

Split the data into a training set and validation set with an 80-20 split. We will not shuffle the data. We use the predefined 10 folds and perform 10-fold cross validation. If we reshuffle the data (e.g. combine the data from all folds and generate a random train/test split) we will be incorrectly placing related samples in both the train and test sets, leading to inflated scores that don't represent our model's performance on unseen data. This will make our results wrong. Our results will NOT be comparable to previous results in the literature, meaning any claims to an improvement on previous research will be invalid. Even if we don't reshuffle the data, evaluating using different splits (e.g. 5-fold cross validation) will mean our results are not comparable to previous research.

We use 10-fold (not 5-fold) cross validation and average the scores. We have seen reports that only provide results for a single train/test split, e.g. train on folds 1-9, test on fold 10 and report a single accuracy score. We strongly advise against this. Instead, perform 10-fold cross validation using the provided folds and report the average score. Not all the splits are as "easy". That is, models tend to obtain much higher scores when trained on folds 1-9 and tested on fold 10, compared to (e.g.) training on folds 2-10 and testing on fold 1. For this reason, it is important to evaluate our model on each of the 10 splits and report the average accuracy.

Model training and evaluation:

I will start with the simple model architecture first (MLP) before training and evaluating it. And use different layers for my models like Dense, Dropout, Activation, Flatten, Convolution2D others if require. My model will be compile on

Loss = categorical_crossentropy

Metrics = accuracy

Optimizer = adam

Will use 100 epochs and batch size will be 32.

Then iterate this process trying different architecture (CNN) and hyper-parameters to reach an accuracy score we are happy with. In CNN I will use layers: Dense, Dropout, Activation, Flatten, Conv2D, MaxPooling2D, GlobalAveragePooling2D, Adam. Filter_size will be 2. Will Train the data in 26 epochs with batch size of 256.

Will create different function for both MLP and CNN model to compare the results on same data.

➔ References

1. Justin Salamon, Christopher Jacoby and Juan Pablo Bello, “Urban Sound Datasets”, “UrbanSound8K”
   https://urbansounddataset.weebly.com/urbansound8k.html
2. Mel-frequency cepstrum Wikipedia page
   https://en.wikipedia.org/wiki/Mel-frequency_cepstrum
3. J. Salamon, C. Jacoby, and J. P. Bello, “A dataset and taxonomy for urban sound research”
   http://www.justinsalamon.com/uploads/4/3/9/4/4394963/salamon_urbansound_acmmm14.pdf