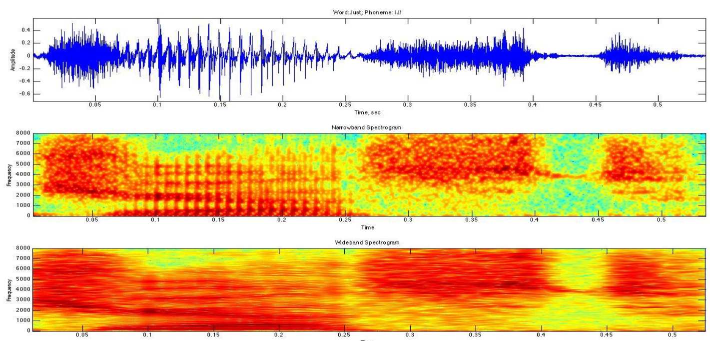
Coding Challenge

Sound Classification using CNN

Suprio Dutta

14th September, 2018



Data Pre-processing

The given dataset is a collection of 2000 audio files containing environmental sounds organized into 50 semantical classes.

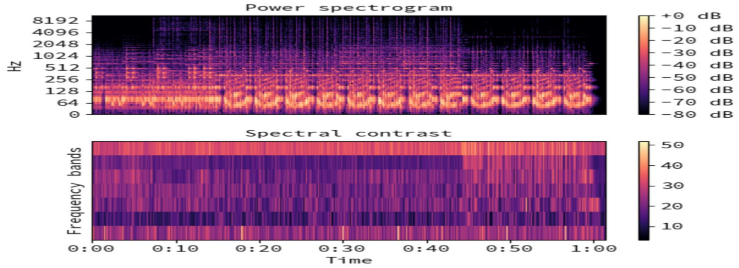
As per the problem statement given I was required to convert the 5 seconds long file to 10 seconds long. For achieving this I performed following steps:

1. I took each files from audio directory and append it to make it a 10 seconds files.
2. Also took the same name as the previous ones and kept it in a different directory.
3. Then I generated spectrum of each file to process it further.

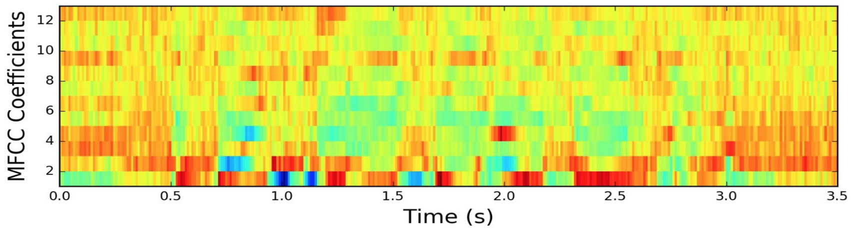
Spectrogram Generation

A spectrogram is a visual representation of the spectrum of frequencies in a sound as they vary with time or some other variable. Spectrograms are sometimes called spectral waterfalls, voiceprints, or voicegrams. Spectrograms are stored as a 2-dimensional numpy arrays. The first axis is frequency and second axis is time.

1. When the data is converted into 10 seconds files, then I generated spectrogram of each file.
2. I created `parse_audio_files` functions to parse each sound file. I used `glob` library to get each file repetitively from the same directory.
3. Then from the `extract_feature` function I extracted some features like mfccs, chroma, mel, contrast, tonnetz.
4. Before extracting, it loaded the files using `librosa.load` method. By default, Librosa's load function will convert the sampling rate to 22.05khz, as well as reducing the number of channels to one, and normalise the data.
5. Then I applied Short-time Fourier Transformation from which I extracted features like contrast & chroma. `chroma_stft` is used to compute a chromagram from a waveform or power spectrogram and `spectral_contrast` is used to compute spectral contrast.



6. Also extracted Mel-frequency cepstral coefficients (mfcc) and mel spectrogram which computes a mel-scaled spectrogram. To obtain MFCCs, a Discrete Cosine Transform (DCT) is applied to the filter banks retaining a number of the resulting coefficients while the rest are discarded.



MFCCs

7. Then I extracted tonnetz to computes the tonal centroid features.
8. I also collected the labels from the files. E.g. one file name is *1-115546-A-48.wav* and 48 is the target class which is *fireworks* class.
9. Later I have done one hot encoding of all the labels where each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1. [Though it was only necessary for categorical variables.]
10. After getting the features in X and labels in y variable I did the split of train and test sets for model training and evaluation.

Model Preparation

Then it comes to model preparation. I tried to build many models but in most of the models accuracy was not good as expected. I also tried to include some dropouts which was also not beneficial. At last I kept the below model.

```
Layer (type)                   Output Shape              Param #
=================================================================
conv1d_5 (Conv1D)             (None, 189, 64)            384
_________________________________________________________________
conv1d_6 (Conv1D)             (None, 185, 64)            20544
_________________________________________________________________
max_pooling1d_3 (MaxPooling1) (None, 37, 64)             0
_________________________________________________________________
conv1d_7 (Conv1D)             (None, 33, 128)            41088
_________________________________________________________________
max_pooling1d_4 (MaxPooling1) (None, 6, 128)             0
_________________________________________________________________
conv1d_8 (Conv1D)             (None, 2, 128)             82048
_________________________________________________________________
global_average_pooling1d_2    (None, 128)                0
_________________________________________________________________
dropout_2 (Dropout)           (None, 128)                0
_________________________________________________________________
dense_2 (Dense)               (None, 50)                 6450
=================================================================
Total params: 150,514
Trainable params: 150,514
Non-trainable params: 0
_________________________________________________________________
```

Here I have used four convolutional layers-

- The first two layers has 64 filters with kernel size of 5,
- The last two layers has 128 filters with kernel size of 5.

In addition, there are two max-pooling layers each of size 5 and one drop out layer (0.5). Then at last I added one fully-connected dense layer with softmax activation function to get the label.

Result

After building the model I fitted it with 2000 epochs which gave me around ~60% accuracy. It took around 560 seconds (9.33 min) to fit.

Then in the last part where you have asked to record a sound and predict it with the model, I recorded a sound taken from basin tap water. But when I give it to the model to predict after doing the processing, it predicted almost accurately. It predicted 'Toilet-Flush' which has a similar sound like this. So, I think the model is performing well with unseen data.