Machine Learning Engineer Nanodegree

Capstone Proposal
Kannappan Natarasan
Apr 17th 2018

I've chosen a current challenge in Kaggle - *Free Sound audio tagging challenge* for this capstone project

https://www.kaggle.com/c/freesound-audio-tagging

Domain Background

Mastering supervised learning on sound clips would be great, and I'm going to apply this to build an app which can monitor sounds from machines and predict the current operating condition as a)running normal b)running with condition A or condition B c) Need a maintenance, thus it can forecast mechanical failures early and avoid accidents.

Deep learning has made greater impact on robotics and automation through computer vision and understand language through text or audio. I've selected current kaggle challenge to classify sound files. When I've explored mechanics of sound and how its stored digitally , figured out that .wav files can be converted into single dimension array. Single dimension array can be used to train CNN and predict label.

As I got very good exposure from MLND , to classify images (3 D array for color images and 2D array for gray scale) through deep learning using MLP or CNN, I'm curious how it would be if I use sound files if train through keras Sequence model.

Citations :

Found this interesting article on speech recognition
https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a
https://www.iotforall.com/tensorflow-sound-classification-machine-learning-applications/
https://www.analyticsvidhya.com/blog/2017/08/audio-voice-processing-deep-learning/

Data Source :
https://www.kaggle.com/c/freesound-audio-tagging/data

Problem Statement

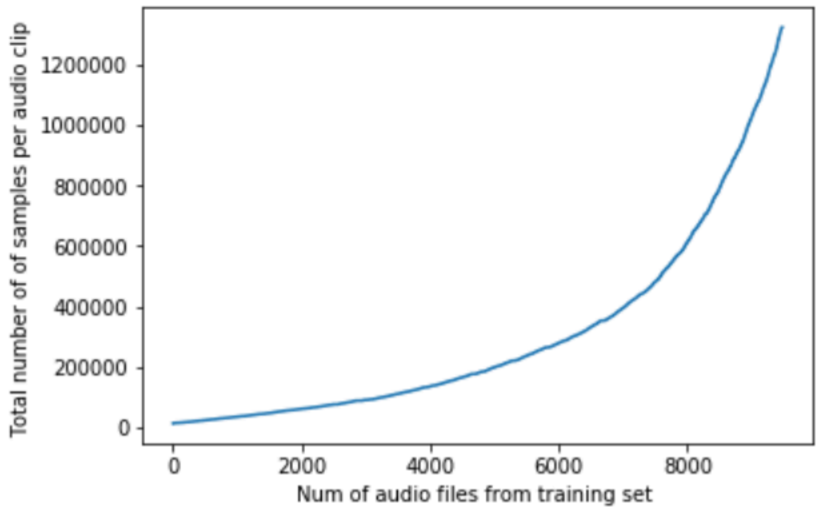
From set of human labeled sound files a model has to be built by training with train dataset in .wav format. Train dataset has sound files which can be converted into 1D array and labels are provided from csv file Train.csv. CNN techniques can be used to train the model. Human labeled data set can be split into train validation and test, so that prediction accuracy can be measured.

Dataset and inputs

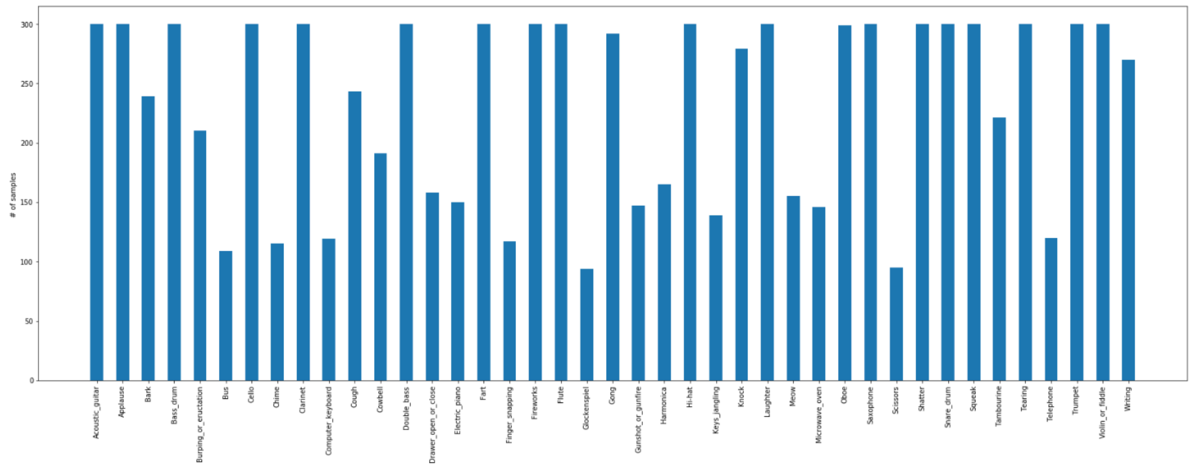
As Kaggle challenges provides cleaned up data set, I can focus only on machine learning problems. In this competition Kaggle provides following datasets.

audio_train.zip : folder containing sound files on .wav format for training
train.csv - which contains the file name and label (files have 41 labels so CNN should have 41 classes)
audio_test.zip : This would be used for kaggle competition to assess contestants model accuracy

Sampling rate in all of these files is standard and equal to 44100, Length of audio is not standar d across files, so I've decide to pad smaller files.



Train files are total to 9473 files with 41 labels.Most of the classes has number of sam ples 300 but minimum goes close to 100. Scissors,Glockenspiel and Bus are least common clas ses, majority of most common classes are music instruments.



Solution statement

Using keras package CNN network can be built to train and classify sound files. Train dataset from Kaggle could be split into train and validation set to measure accuracy of the prediction. By fine tuning hyper parameters the model for better accuracy could be arrived.

Benchmark model

I'll use default method used in keras Sequence to predict before training to benchmark this model. I'm planning to use Conv1D model from keras package . For benchmark I would use few layers.

Evaluation metrics

I'll use Mean Average Precision@3 , thus the predication accuracy can be calculated based on first three predicted files.

$$MAP@3 = \frac{1}{U} \sum_{u=1}^{U} \sum_{k=1}^{min(n,3)} P(k)$$

U	:	the number of scored audio files in the test data,
P(k)	:	the precision at cutoff k
n	:	the number predictions per audio file.

https://www.kaggle.com/wendykan/map-k-demo

Project design

1. Data preparation
    1. Download required dataset
    2. Install packages which can read and process .wav files
2. Exploratory analysis to understand data and
    1. Visualize data
    2. Provision to play sound
3. Preprocessing
    1. Length of audio file for training has to be standardized
        1. Either smallest length of all files could be used
        2. Smallest file is very small, a standard audio length has to be used for smaller files padding has to be applied.
    2. Look for outliers
    3. Normalization
    4. Split data into train and test
4. Modeling
    1. Design CNN architecture with following layers
        1. Conv1D
        2. MaxPooling

    3. dropout
2. Train model
3. Evaluate model
4. Fine Tune hyper parameters