Machine Learning Engineer Nanodegree

Capstone Proposal

Jari-Pekka Ryynänen June 24, 2017

Proposal

Domain Background

Audio event classification is a task of giving appropriate label to audio. Having automatic system for labeling has several easy-to-understand applications, e.g:

- categorising audios/videos and making semantic search possible
- displaying extra information with captions (very helpful for deaf people)
- triggering some actions based on the audio event (like saving only speech)

There are several papers tackling this problem [1,2,3]. As the amount of data available has grown, the deep learning methods have become more suitable also for this problem.

Problem Statement

Problem to solve: Create a system that is capable of giving single label to a short audio segment. More precisily, given audio sample, predict a label that describes that sample, where label is from predefined set. The approach will be supervised, i.e. system is trained to do this prediction by showing sample-label pairs. Precise predictions with coarse labels are preferred over imprecise with detailed labels.

Datasets and Inputs

AudioSet: Large-scale collection (over 2M samples) of human-labeled 10-second sound clips drawn from YouTube videos [4]. This dataset is perhaps the largest freely available one, and has very good documentation and support. There is also starter code which is helpful when starting to work with the dataset [5].

AudioSet includes balanced subsets for training and evaluation. AudioSet samples might have multiple labels assigned to each sample; our target is however to predict single label.

There are also precalculated audio features (128-dimensional @ 1 Hz). Unfortunately the feature extraction process used for those is not provided anywhere at the detail that would allow one to reproduce the extraction.

Dataset is described in more detail in [4] and [6].

Solution Statement

Intead of using the whole dataset, which is huge, balanced training and balanced evaluation sets are used, both containing more than 20k samples. Target labels will be the top-level labels in AudioSet ontology: “Human sounds”, “Source-ambiguous sounds”, “Animal”, “Sounds of things”, “Music”, “Natural sounds”, “Channel, environment and background”. These labels are used instead of the exact ones to simplify the problem.

Rather than providing raw audio for classifier, frequency domain content is extracted for short frames using Mel-frequency cepstrum coeffiecients (MFCC) [7,8]. Using those as inputs, recurrent neural network is trained to do the classification. Recurrent neural networks are suitable for working with continuous audio, as they are able to capture also temporal features. Evaluation set is kept aside until the classifier is in its final form (before that, model generalization is approximated with splits done within the training set).

Benchmark Model

In [6] benchmark model is reported with “balanced mean Average Precision across the 485 categories of 0.314”. However, this result is not directly comparable with our target, as we will use more coarse labels, and we will not do multi-label classificiation.

In [2] are results of audio scene classification with F1-score: “approach obtains an F1-score of 97.7%”. The audio scene classification has subtle differences to our task at hand, but eventually it is about classifying audio segments, so it makes sense to compare our results with it. The dataset used in [2] is LITIS-Rouen, which has 19 categories.

In this work, simple logistic regression model will be trained without parameter tuning to give baseline. Also zero-hypothesis (model that predicts first category always) is used to gain understanding how well our model is performing.

Evaluation Metrics

Balanced error measure which is easy to understand, and is comparable with values from [2] is **F1-score**: Harmonic mean of precision and recall. Mathematical equation for F1 is *2*precision*recall/(precision+recall).*

Project Design

The main tools will be Tensorflow and Python.

The first steps will be about the preparation of the data. Using the ontology provided in [4], the top-level labels will be assigned to data, and samples that will still after this have multiple labels will be duplicated (e.g. if one sample has three distinct labels, it will be turn in to three samples which each has one of the labels).

After the relevant audio segments are downloaded with youtube-dl [9], feature extraction can be done with [8]. Extracted features can then be fed to classifier.

There are plenty of tunable properties in both feature extractor and in classifier, and it is very likely that adjusting them will take the most time.

Zero-hypothesis model can be obtained already at the very beginning. Baseline logistic regression has to be re-trained each time feature extraction is modified.

At the point where the model does not seem to improve anymore by just parameter tuning, it will be considered to be as good as it gets within this project and the evaluation set is used to get the final score.

References

1. Kons, Z., Toledo-Ronen, O.: Audio Event Classification Using Deep Neural Networks, 2013 https://pdfs.semanticscholar.org/1881/acfe27135da6c717fb770dfa3793b8b225d5.pdf
2. Phan, H. et al.: Audio Scene Classification with Deep Recurrent Neural Networks, 2017 https://arxiv.org/pdf/1703.04770.pdf
3. Bae, H. et al: Acoustic Scene Classification Using Parallel Combination of LSTM and CNN, 2016 https://www.cs.tut.fi/sgn/arg/dcase2016/documents/challenge_technical_reports/
4. AudioSet: A large-scale dataset of manually annotated audio events https://research.google.com/audioset/dataset/index.html
5. YouTube-8M Tensorflow Starter Code: https://github.com/google/youtube-8m
6. Gemmeke, J. et al: Audio Set: An ontology and human-labeled dataset for audio events, 2017 https://research.google.com/pubs/pub45857.html
7. Mel-frequency cepstrum, https://en.wikipedia.org/wiki/Mel-frequency_cepstrum
8. python_speech_features https://github.com/jameslyons/python_speech_features
9. youtube-dl https://github.com/rg3/youtube-dl