Machine Learning Engineer Nanodegree Capstone Proposal

Mike Fuller, 12 March 2020

Proposal

Domain Background

The sub-field of Machine Learning known as Machine Listening is a burgeoning area of research using signal processing for the automatic extraction of information from sound by a computational analysis of audio. There are many different areas of research within this field as demonstrated by the latest Detection and Classification of Acoustic Scenes and Events (DCASE) 2020 Challenge, a machine learning challenge dedicated to the research and development of new methods and algorithms. These include:

- Acoustic Scene Classification
- Sound Event Detection and Localization
- Sound Event Detection and Separation in Domestic Environments
- Urban Sound tagging
- Automated Audio Captioning

I am personally extremely intrigued by this new field due to having studied Environmental Acoustics as my Master's of Science and since working as professional acoustic consultant for over 8 years. Recent developments in machine learning algorithms have allowed significant progress to be made within this area with the potential applications of the technology being wide and varied and could prove an extremely useful tool for the acoustic practitioner. I am further excited by the potential applications of this within acoustic ecology for the monitoring and assessment of ecosystem studies. I am currently working on my own bird sound classifier for the Chingaza National Natural Park in Colombia as a personal project so this is of great interest.

Problem Statement

The Freesound Audio Tagging 2019 Kaggle Competition provides a basis for my research project[1].

The challenge is to develop a system for the automatic classification of multi-labelled audio files within 80 categories, that could potentially be used for automatic audio/video file tagging and/or real-time sound event detection with noisy or untagged audio data. This has historically been investigated in a variety of ways:

- Conversion of audio to mel-spectrogram images fed into CNNs
- End-to-End Deep learning
- Custom architectures involving auto-encoders
- Features representation transfer learning with custom architectures and Google's Audioset

In addition, the problem of the classification of weakly labelled data from large-scale crowdsourced datasets provides a further problem for investigation [2]. The problem is clearly quantifiable in that a number of accuracy metrics could be used to quantify the accuracy of the model's predictions.

Datasets and Inputs

The dataset used in the challenge is called FSDKaggle2019 [1] and was collected by members of Freesound (a Creative Commons Licensed sound database from the Music Technology Group of Universitat Pempeu Fabra, Barcelona) and Google Research's Machine Perception Team [2].

The dataset comprises audio clips from the following existing datasets:

- Freesound Dataset (FSD): a dataset being collected at the MTG-UPF based on Freesound content organized with the AudioSet Ontology and manually labelled by humans.
- The soundtracks of a pool of Flickr videos taken from the Yahoo Flickr Creative Commons 100M dataset (YFCC) which are automatically labelled using metadata from the original Flickr clips. These items therefore have significantly more label noise than the Freesound Dataset items.

The data comprises 80 categories labelled according to Google's Audioset Ontology [3] with ground truth labels provided at the clip level. The clips range in duration between 0.3 to 30s in uncompressed PCM 16 bit, 44.1 kHz mono audio files.

Solution Statement

The proposed solution is to first download the data into an AWS S3 bucket. Next (as detailed further below in the Project Design section) verify the sample rates, bit-rates and channels of the audio files and then convert them to log-mel spectrograms which can be fed into a CNN multi-label classifier. Transfer learning from an ImageNet trained model will be used to improve the accuracy.

Benchmark Model

There are many different potential neural network architectures for this problem as described above, however, the Winner of the Kaggle competition achieved a lwl-wrap score of 75.98% which sets the state of the art performance and is available on GitHub [1]. The Pytorch model follows a similar architecture to the other high performing submissions, incorporating log-mel-spectrograms fed into a CNN model with skip connections, however, custom attention layers and auxiliary classifiers were also incorporated. Augmentation of the dataset was used (SpecAugment and Mixup). In addition hand engineering including relabelling of the dataset examples with a low score was used.

Evaluation Metrics

The original Kaggle competition used label-weighted label-ranking average precision (a.k.a lwlrap) as the evaluation metric. The label-ranking average precision algorithm is described in detail within the sklearn implementation[1]. The metric provides the average precision of predicting a ranked list of relevant labels per clip. The overall score is the average over all the labels in the test set, with each label having equal weight (rather than equal weight per test item), as indicated by the "label-weighted" prefix. This is defined as follows[2]:

$$lwlrap = \frac{1}{\sum_s |C(s)|} \sum_a \sum_{e\epsilon\ C(s)} Prec(s,c)$$

where $Prec(s,c)$ is the label-ranking precision for the list of labels up to class c and the set of ground-truth classes for sample s is $C(s)$. $|C(s)|$ is the number of true class labels for sample s.

The Kaggle competition provides a Google Colab example implementation [2]

Project Design

The proposed pipeline for the solution will be as follows and will be undertaken within Jupyter Notebooks (using Python), the Fastai library and Pytorch and AWS:

1. Exploratory Data Analysis - download the dataset to AWS S3 and view the labels, sample rates, bit-rates, lengths, channels (mono/stereo) for each file in order to direct initial signal processing stage and approach towards the dataset splitting.
2. Data Preparation - depending on the outcome of the EDA, this signal processing stage will likely involve converting the data to tensors and then resampling (to ensure uniform sample rates/bit-rates), downmixing (from stereo to mono), trimming (to ensure uniform duration) in order to allow the uniform tensor data to be converted to log-mel-spectrogram representations of the audio. A log-mel-spectrogram is a spectrogram representation of the audio (i.e. a frequency-domain representation based on the Fourier Transform, with x-axis = time, y axis = frequency and colour depth = relative sound intensity), which has been has been converted to the Mel scale on the y-axis by a non-linear transform, in order to be more representative of the highly non-linear magnitude and frequency sensitivity of the human ear[1].

   The data will further be split into training, validation and test sets as appropriate. In the initial model development and data augmentation stage a subset of the data will be used to allow for fast iteration and evaluation of the model.
3. Model development - it is proposed that a pretrained Convolutional Neural Network will be used for the analysis and inference of the log-mel-spectrograms using a transfer-learning approach in order to achieve the highest accuracy. The network will have its base model frozen (other than the batch-norm layers) and the custom head trained first to a high level of accuracy. The base model will then be unfrozen and trained further in a fine-tuning stage to further improve the accuracy.
4. The final aim would be to create an endpoint and deployed web app such that users can upload an audio file for inference, with the model returning its prediction.

---

1. https://www.kaggle.com/c/freesound-audio-tagging-2019/overview ↩ ↩ ↩ ↩ ↩
2. Learning Sound Event Classifiers from Web Audio with Noisy Labels - Fonseca et al. 2019 https://arxiv.org/abs/1901.01189 ↩ ↩ ↩ ↩
3. https://research.google.com/audioset///////ontology/index.html ↩