fma-selection

This project aims to use techniques in optimization for feature selection. The goal of this project is to expand on the research of Defferrard et. al, and apply the techniques for music analysis.

FMA: A Dataset For Music Analysis

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, Xavier Bresson, EPFL LTS2.

The dataset is a dump of the Free Music Archive (FMA), an interactive library of high-quality, legal audio downloads. The referenced paper is a **pre-publication release**.

Data

All metadata and features for all tracks are distributed in **fma_metadata.zip** (342 MiB). The below tables can be used with pandas or any other data analysis tool. See the paper or the [usage] notebook for a description. * `tracks.csv`: per track metadata such as ID, title, artist, genres, tags and play counts, for all 106,574 tracks. * `genres.csv`: all 163 genre IDs with their name and parent (used to infer the genre hierarchy and top-level genres). * `features.csv`: common features extracted with librosa. * `echonest.csv`: audio features provided by Echonest (now Spotify) for a subset of 13,129 tracks.

Then, you got various sizes of MP3-encoded audio data:

1. **fma_small.zip**: 8,000 tracks of 30s, 8 balanced genres (GTZAN-like) (7.2 GiB)
2. **fma_medium.zip**: 25,000 tracks of 30s, 16 unbalanced genres (22 GiB)

Prerequisites

1. Download some data, verify its integrity, and uncompress the archives.`curl -O`
   ```
   https://os.unil.cloud.switch.ch/fma/fma_metadata.zip
   curl -O https://os.unil.cloud.switch.ch/fma/fma_small.zip
   curl -O https://os.unil.cloud.switch.ch/fma/fma_medium.zip


   echo "f0df49ffe5f2a6008d7dc83c6915b31835dfe733  fma_metadata.zip" | sha1sum -c -
   echo "ade154f733639d52e35e32f5593efe5be76c6d70  fma_small.zip"    | sha1sum -c -
   echo "c67b69ea232021025fca9231fc1c7c1a063ab50b  fma_medium.zip"   | sha1sum -c -

   unzip fma_metadata.zip
   unzip fma_small.zip
   unzip fma_medium.zip
   ```
2. Clone the repository. `git clone https://github.com/hfloresr/fma-selection.git`
   ```
   cd fma-selection
   ```
3. Install the Python dependencies from `environment.yml`. Depending on your usage, you may need to install ffmpeg or graphviz. Install CUDA if you want to train neural networks on GPUs (see Tensorflow's instructions). `conda env create --file environment.yml`
4. Fill in the configuration. ```sh cat .env AUDIO_DIR=/path/to/audio