Project (we are updating it regularly)

EE698V – Machine Learning for Signal Processing

Introduction

For the rest of the course, we will be having a hands-on project. You can do it individually or form a team of up to 2 students for the project. This project will serve as a part of end-sem. The other part will be either an MCQ or a coding exam.

There are two tasks given below – both will be evaluated. You can build your system and train it using the data given here. The evaluation will be done on a test set that will be released later around end sem. Students doing the best on test set will get more marks.

Final submission consists of at least the following:

- task1_labels_test.csv
- task2_labels_test.csv
- task1_code.ipynb
- task2_code.ipynb
- report.pdf

It is an audio classification problem, technically known as "Audio Event Detection". The content is available here:
https://iitk-my.sharepoint.com/:f:/g/personal/vipular_iitk_ac_in/EqibR9AbEi9GiPeGGOZUeVsBn78UyhMHVF0hL_yAl8IxAg?e=c7upzH
See README.docx for description.

Each audio file corresponds to an event class, e.g., children-playing, dog-barking, drilling, etc. The folder "audio_train/" contains the audio files and the file "labels_train.csv" contains the class labels.

TASK 1

Given an audio file corresponds to a single event, find out that event.

Task 1 example: street_music (right click to Play)

Evaluation

Accuracy

TASK 2

Given an audio file contains a sequence of events occurring one after the other, find out that sequence of events. A sequence can contain at least 1 and at most 5 events.

Task 2 example: street_music, dog_bark, engine_idling (right click to Play)

Evaluation

Edit distance

- Return sequence of classes separated by hyphen "-".
    - E.g. street_music-dog_bark-engine_idling
- Labels would not repeat consecutively, e.g., street_music-dog_bark-dog_bark will be labeled as street_music-dog_bark

Competition Rules

- No extra/external data allowed for training. You can use only the training data provided.
- Allowed libraries:
    - For data processing: Numpy, pandas
    - General libraries: glob, pdb, string, os, sys,
    - For ML: Scipy, Scikit-learn, Levenshtein
    - For audio processing/feature extraction: librosa
    - For deep learning: Keras, tensorflow, pytorch
    - For HMMs: https://github.com/hmmlearn/hmmlearn and https://github.com/larsmans/seqlearn
    - Let the instructor know if you want to use any other library
- The test set will not contain audio (wav) files but spectrograms. The feature extraction function is available in "shared_train/utils.ipynb".

- The test set will be released for a limited time. You have to run your codes to predict the labels and submit in csv files. The format will be shared soon.
- Grading will be done based on evaluation results, the approach used and the quality of report (presentation). Weightage: task 1 results (30%), task 2 results (30%), report and codes (40%).

Codes and Format

On the day of exam, we will release "feats/*.npy" for both the tasks, as given in "sample_test_task*/".

- Functions in utils.py are used to generate "feats/*.npy"

- For evaluation:
  Your predicted labels (est.csv) should be in the same format as labels.csv
  We will use eval_model.py to get your score.

  E.g.,

  ```
  $ python eval_model.py sample_test_task1/labels.csv
  est.csv 1
  Your score is: 90/100

  $ python eval_model.py sample_test_task2/labels.csv
  est.csv 2
  Your score is: 97/100
  ```

Useful Resources

- (educational) AED talk by Justin Salamon
  https://www.youtube.com/watch?v=zvccOFz2KxI&ab_channel=SpeechandAudiointheNortheast%28SANE%29
- (interesting) Application of AED in Amazon Alexa:
  https://www.youtube.com/watch?v=-nKelNVVblM&ab_channel=Amazonre%3AMARS
-