EE599 Deep Learning – Homework 5

March 26, 2019

1 Objective

Spoken Language Identification (LID) is broadly defined as recognizing the language of a given speech utterance [1]. It has numerous applications in automated language and speech recognition, multilingual machine translations, speech-to-speech translations, and emergency call routing. In this homework, we will try to classify three languages (English, Hindi and Mandarin) from the spoken utterances that have been crowd-sourced from the class.

2 Goals

In this homework you will learn to:

1. Extract Mel-frequency cepstral coefficients (MFCCs) from audio, which will be employed as features.
2. Implement a GRU/LSTM model, and train it to classify the languages.

3 Steps

3.1 Dataset

The dataset has a bunch of wav files and a csv file containing labels. The wav file names are anonymized, and class labels are provided as integers. You are supposed to train with the provided integer class labels.

The following mapping is used to convert language IDs to integer labels:

```
mapping = dict{'english': 0, 'hindi': 1, 'mandarin': 2}
```

3.2 Audio format

The wav files have 16KHz sampling rate, single channel, and 16-bit Signed Integer PCM encoding.

3.3 Voice Activity Detection (VAD)

You can do simple energy based VAD using **sox** as described in discussions, but be careful about the energy threshold since it can chop beginning and end parts of a word. You are encouraged to try both with and without energy based VAD, and compare the performances

3.4 Data split

We provide a training set after holding out a non-overlapping test set for evaluation. The test and training sets have no speakers in common, so that we can *truly* verify if the DNN can recognize language independent of speakers. You are free to create any validation split from the training set.

3.5 MFCC Features

MFCC features are widely employed in various speech processing applications including LID [2]. You are supposed to employ MFCC features for this homework, so that the input features are same for everyone and the differences in performance do not depend on the choice of features.

You can utilize **Librosa** (`https://librosa.github.io/librosa/index.html`) to extract 64 dimensional MFCC features for all utterances. A sample code snippet is provided below:

```
import librosa
y, sr = librosa.load('audio.wav',sr=16000)
#sr should return 16000, y returns the samples
mat = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=64, n_fft=int(sr*0.025), hop_length=
    int(sr*0.010))
print(y.shape, sr, mat.shape)
```

A detailed overview of MFCC features and librosa usage will be given in discussion 11.

3.6 Sample length

The full audio files are $\sim$ 10 minutes long. It might be too long to train an RNN. You can create multiple 3-10 seconds samples from every utterance and assign them the same *label* as the original utterance. The choice of sequence length is up to you, and you are encouraged to experiment with that. You can follow existing literature such as [1, 2]. Please note that the *held out* test set might have short and as well as long utterances.

3.7 Submission Material

You will submit a "streaming model" based on your training. The conversion for the sequence-trained model and the streaming model can be done as shown in lecture (see `mind_reader.py`). You streaming model should give three output scores (probabilities) for each 10 msec feature vector that it accepts – i.e., there are the probability of English, Hindi, and Mandarin. In doing your own self-evaluation, you should plot these scores as a function of time for sample input files.

4 Evaluation

Classification accuracy will be the primary metric for test set evaluation. The classification decision will be based on the average of the probabilities over time.

5 Submission

Please keep an eye on Piazza for submission related information. The above is a guideline and the specifics of the submission format and materials will be provided as the date approaches.

References

[1] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. J. Moreno, "Automatic language identification using long short-term memory recurrent neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[2] A. Lozano-Diez, O. Plchot, P. Matejka, and J. Gonzalez-Rodriguez, "Dnn based embeddings for language recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5184–5188.