Speech Processing and Recognition - Exercise 2

Due Date: May 7th 2019 (there will be no extensions!)

Yossi Adi, Joseph Keshet and Felix Kreuk

April 16, 2019

1 DTW

In this exercise you will implement your first word recognizer using the Dynamic Time Warping (DTW) algorithm. Your algorithm will recognize the digits 1 - 5.
For that, You will implement the DTW algorithm as described below, where the distance metric (d) should be the Euclidian distance.

$$\begin{aligned} DTW[0,0] &= d(0,0) \\ DTW[i,j] &= d(i,j) + \min \begin{pmatrix} DTW[i-1,j-1] \\ DTW[i,j-1] \\ DTW[i-1,j] \end{pmatrix} \end{aligned} \tag{1}$$

1.1 Dataset

You are provided with five labeled examples for each class ['one', 'two', 'three', 'four', 'five'] to use as your training set. For the test set you are provided with 250 unlabeled examples. Each file is exactly 1 second long.

1.2 Code

In practice, DTW can be applied to varying length sequences. However, to better understand the advantages of the DTW algorithm, you will compare its performance to a standard Euclidian distance (recall each file is exactly 1 second long).
To sum up, you need to run a 1 nearest neighbor classifier using both Euclidian distance and DTW distance. You need to compute both distance metrics (Euclidian and DTW) for each file in the test set with all training examples. Then, classify each test file as the label of the file with the minimal distance.
You should generate a file named: ‘output.txt‘, with the predictions for each test file using both euclidian distance and DTW distance. The output file should be constructed as follows:

<filename> - <prediction using euclidian distance> - <prediction using DTW distance>

For example,

```
f6581345_nohash_1.wav - 1 - 3
fb7c9b3b_nohash_0.wav - 2 - 1
fb9d6d23_nohash_0.wav - 4 - 2
fda46b78_nohash_2.wav - 1 - 1
...
```

1.3 Features

In class, we learned how to transform a time domain waveform to the frequency domain using the Fourier Transform. However, in many applications we would like to use more compressed representation. One representation like that is the Mel Frequency Cepstrum Coefficients (MFCCs). In order to extract these features and load the wave files, you will use a python package called 'librosa', using the following lines of code:

```python
import librosa
y, sr = librosa.load(f_path, sr=None)
mfcc = librosa.feature.mfcc(y=y, sr=sr)
```

The dimensions of the MFCC object should be (20, 32), meaning 20 MFCC features over 32 time steps. A more detailed explanation of the MFCC can be found here: [paper], and an intuitive explanation can be found here: [blog].

2 What to submit?

You should submit the following files:

- A `txt` file, named `details.txt` with your name and ID.
- Python 3.6 code with your implementation.
- output.txt file with your prediction per examples, using both euclidian distance and DTW distance, as described in 1.2.
- Part of your grade will consist of automatic checks using the `Submit` system. Make sure you can access it!