



تمرین (۱)

نام و نام خانوادگی:

مهتاب مهدوی

شماره دانشجویی:


۹۸۴۴۰۲۲۳

استاد محترم:

دکتر بهزاد بختیاری

مقدمه

پیاده‌سازی این پروژه در محیط Jupyter Lab و با استفاده از کتابخانه‌های زیر بوده است:

librosa 
numpy 
dtw 
lpython.diplay 
matplotlib.pyplot 
sklearn.preprocessing 
pandas 
math 

با استفاده از تابع load موجود در کتابخانه librosa، ۱۰ فایل صدا حاوی نام‌خواندگی خود را خوانده و سپس ذخیره می‌کنیم. این کار را برای فایل‌های صوتی با صدای افراد دیگر نیز انجام می‌دهیم. تکه کد زیر، تابع خواندن این فایل‌ها را نمایش می‌دهد.

```
def load_voices(path):  
    voices = []  
    sampling_rate = []  
  
    files = librosa.util.find_files(path, ext = ['wav'])  
    files = np.asarray(files)  
  
    for i, file in enumerate(files):  
        y, sr = librosa.load(file, mono = True)  
        voices.append(y)  
        sampling_rate.append(sr)  
  
    return voices, sampling_rate
```

حذف سکوت

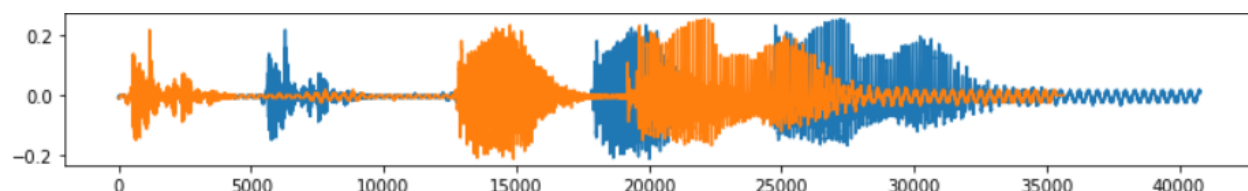
در این مرحله به کمک تکه کد زیر قسمت‌های حاوی نویز و سکوت از سیگنال صداها حذف می‌کنیم.

```
def remove_silence(voices):  
    signals = []  
    for i, voice in enumerate(voices):  
        yt, index = librosa.effects.trim(voice, top_db = 20)  
        y_harm, y_perc = librosa.effects.hpss(yt)  
        signals.append(y_harm + y_perc)  
  
    return signals
```

سیگنال زیر نمونه‌ای نمایش داده شده از سیگنال صدا قبل حذف سکوت (رنگ آبی) و پس از حذف آن (رنگ نارنجی) است. همان‌طور که مشاهده می‌کنید، طول سیگنال کاهش یافته‌است. با قطعه کد زیر می‌توان سیگنال صدا را پلات نمود.

```
def draw_signals(voices, sampling_rate):
    for i, voice in enumerate(voices):
        figure = plt.gcf()
        figure.set_size_inches(12.5, 19.5)

        plt.subplot(10, 1, i + 1)
        plt.plot(voice)
        IPython.display.Audio(data = voice, rate = sampling_rate[i])
```



فریم‌بندی و پنجره‌گذاری

این کار را با استفاده از ۲ تابع `util.frame` و `normalize` که برای `librosa` و دومی برای `sklearn.preprocessing` انجام می‌شود.

```
def frame_and_normalize(signals):
    normal_signals = []

    for signal in signals:
        signal = librosa.util.frame(signal, frame_length = 2048, hop_length = 64)
        signal = normalize(signal, axis = 0)

        normal_signals.append(signal)

    return normal_signals
```

استخراج ویژگی

در این قسمت با استفاده از کتابخانه `librosa` و توابع مختلف آن ویژگی‌های سیگنال صدا را استخراج می‌کنیم. ویژگی‌های مورد استفاده در این بخش انرژی، میانگین، نرخ عبور از صفر و mfcc هستند. پس از استخراج این ویژگی‌ها را در ماتریس ویژگی ذخیره می‌کنیم.

```
def feature_extraction(normal_signals):
    matrix_list_of_features = []
    frames = []

    for signal in normal_signals:
        matrix_features = []
        frames = []
        frames = signal

        for frame in frames:
            energy = np.asarray([value * value for value in frame])
            mean = np.mean(frame)
            zcr = np.ndarray.flatten(librosa.feature.zero_crossing_rate(frame))[0]
            mfcc = np.ndarray.flatten(librosa.feature.mfcc(y = frame[:1], sr = 2048))[0]

            matrix_features.append(energy[0])
            matrix_features.append(mean)
            matrix_features.append(zcr)
            matrix_features.append(mfcc)

        matrix_list_of_features.append(np.asarray(matrix_features))

    return matrix_list_of_features
```

DTW

فاصله هر سطر از ماتریس ویژگی که از قسمت قبل به دست آمده (هر سطر برای یک سیگنال صدا) را با خود آن و سایر سطرها به کمک dtw به دست آورده و در ماتریس فاصله ذخیره می‌کنیم.

```
def calcute_dtw(matrix_features):
    matrix_distance = np.zeros((len(matrix_features), len(matrix_features)))
    avg = []

    for i in range(0, len(matrix_features)):
        for j in range(i + 1, len(matrix_features)):
            sequ1 = matrix_features[i]
            sequ2 = matrix_features[j]

            dist = dtw(sequ1, sequ2)
            matrix_distance[i][j] = dist.distance
            matrix_distance[j][i] = matrix_distance[i][j]

    for i in range(0, len(matrix_features)):
        avg.append(matrix_distance[:, i].mean())

    return matrix_distance.mean(), avg, matrix_distance
```

سپس نتایج را به شکل جدول و با کمک تابع زیر نمایش می‌دهیم:

```
def print_results(matrix_distance):
    values = []

    for i in range(len(matrix_distance)):
        values.append(f"voice {i + 1}")

    df = pd.DataFrame(matrix_distance, columns = values)
    print(df)
```

	voice 1	voice 2	voice 3	voice 4	voice 5 \
0	0.000000	392593.045936	464393.580084	351792.794535	474107.914272
1	392593.045936	0.000000	393469.859802	350536.566460	554326.456997
2	464393.580084	393469.859802	0.000000	297050.801508	427057.747595
3	351792.794535	350536.566460	297050.801508	0.000000	375300.560219
4	474107.914272	554326.456997	427057.747595	375300.560219	0.000000
5	372152.908373	312267.071085	294659.667303	189726.524763	440799.319202
6	449047.387380	359297.627369	348502.447676	334896.319204	535988.022503
7	389349.454572	373352.376519	251279.956234	185504.340490	371394.567838
8	321451.497392	366198.854239	364245.744675	283008.770411	485571.775106
9	391207.620225	347950.285343	264648.794162	175443.017744	391358.069602

	voice 6	voice 7	voice 8	voice 9	voice 10
0	372152.908373	449047.387380	389349.454572	321451.497392	391207.620225
1	312267.071085	359297.627369	373352.376519	366198.854239	347950.285343
2	294659.667303	348502.447676	251279.956234	364245.744675	264648.794162
3	189726.524763	334896.319204	185504.340490	283008.770411	175443.017744
4	440799.319202	535988.022503	371394.567838	485571.775106	391358.069602
5	0.000000	280320.761034	211189.534802	323292.867400	207825.836679
6	280320.761034	0.000000	361644.941881	412109.059182	346121.389112
7	211189.534802	361644.941881	0.000000	312030.679127	165699.970900
8	323292.867400	412109.059182	312030.679127	0.000000	318694.149785
9	207825.836679	346121.389112	165699.970900	318694.149785	0.000000

تصدیق هویت

در این بخش ابتدا سطح آستانه مناسب را پیدا کرده و ویس مرجع را با توجه به این موضوع انتخاب می‌کنیم. سپس با استفاده از dtw فاصله ماتریس ویژگی این ویس‌ها را با ویس مرجع به‌دست آورده و مقایسه می‌کنیم. همان‌طور که می‌بینید این صداها پس از مقایسه، تصدیق نشده‌اند.

```
[17]: verification(avg, mean, matrix_features, matrix_features_test)
```

```
Not Verified
Not Verified
Not Verified
Not Verified
Not Verified
Not Verified
```

```
def verification(avg, mean, matrix_features, matrix_features_test):  
    min = inf  
    index = 0  
  
    for i in range(len(avg)):  
        if abs(avg[i] - mean) < min:  
            index = i  
            min = abs(avg[i] - mean)  
  
    reference = matrix_features[index]  
  
    for test in matrix_features_test:  
        dist = dtw(test, reference)  
  
        if mean > dist.distance:  
            print("Verified")  
        else:  
            print("Not Verified")
```

کد این پروژه بر روی گیت‌هاب با لینک زیر قرار گرفته است:

<https://github.com/mahtabmahdavi/Speech-Processing.git>