تکلیف چهارم درس پردازش گفتار

دانشکده‌ی مهندسی
کامپیوتر دانشگاه
اصفهان

استاد درس:
دکتر حمیدرضا برادران

دانشجو:
امیررضا صدیقین
۹۹۳۶۱۴۰۲۴

---

۱. در این تمرین شما با یک مجموعه شامل ۲۰ گوینده (۱۰ گوینده مرد و ۱۰ گوینده زن) کار می‌کنید. هر گوینده ۱۰ فایل صوتی دارد که ۷ فایل آن به عنوان مجموعه یادگیری برای آن گوینده و ۳ فایل به عنوان مجموعه تست برای آن گوینده در نظر گرفته شده است.

در بخش Preprocess functions توابع زیر تعریف شده است:

- Read-voice: تابعی برای خواندن یک فایل صوتی.
- Pre_emphasis : عملیات پیش تاکید در پیش پردازش.
- Framming: عملیات فریم‌بندی برای یک فایل صوتی.
- Windowing: تابعی برای پنجره‌گذاری در یک فریم.
- Preprocess: تابع پیش‌پردازش بر روی داده‌ی صوتی استخراج شده. (پیش‌تاکید، فریم‌بندی و پنجره‌گذاری.)

۲. ابتدا با استفاده از الگوریتم VAD پیاده‌سازی شده در تمارین قبلی، بخش‌های سکوت را از فایل‌های صوتی از بین ببرید.

در بخش VAD function تابعی تحت عنوان VAD وجود دارد که تعدادی فریم در ورودی گرفته و بر اساس لگاریتم انرژی و تعیین یک آستانه (مقدار 6-)، فریم‌های سکوت را حذف می‌کند.

۳. ویژگی‌های MFCC و LPC مربوط به تمام فایل‌های صوتی مجموعه یادگیری و تست را ایجاد کنید. توجه کنید که فایل‌های مجموعه داده، 8 kHz هستند. برای به دست آوردن ضرایب MFCC و LPC، از روش‌هایی که در تمارین قبل پیاده‌سازی کرده‌اید استفاده کنید. برای ویژگی‌های MFCC، در نهایت با اضافه کردن مشتقات اول و دوم، هر بردار MFCC باید ۳۹ بعدی باشد.

در بخش LPC and MFCC features functions با استفاده از کتابخانه‌های spectrum و librosa، ویژگی‌های LPC و MFCC برای هر فریم بدست می‌آید. (توابع get_lpc، get_mfcc). همچنین دو تابع lpc_feature_of_frames و mfcc_feature_of_frames به صورت بهینه، ویژگی‌های مورد نظر را برای فریم‌های استخراج شده، بدست می‌آورد. تعداد بعد ویژگی‌های LPC ۳۹ تا، ویژگی‌های MFCC (شامل خود ضرایب و مشتقات درجه یک و دو) نیز دارای ۳۹ بعد می‌باشد.

در بخش read data and make dataframe نیز مسیر فایل‌های موجود شناسایی و خوانده شده است و برای هر صوت خوانده شده، پیش‌پردازش (pre-emphasis، framing، windowing و VAD) انجام گرفته، سپس برای هر فایل ویژگی‌های ذکر شده در بخش قبل، استخراج شده است. در انتها دو دیتافریم train_df و test_df بدست آورده شد. هر کدام این دیتافریم‌ها شامل speaker، data، آرایه‌ی lpc و آرایه‌ی mfcc است.

۴. برای هر گوینده یک مدل GMM بر مبنای ویژگی‌های MFCC و یک مدل GMM بر مبنای ویژگی‌های LPC آموزش دهید. مدل‌های GMM ذکر شده باید ۳۲ مولفه‌ای با کواریانس‌های قطری باشد. برای مدل GMM می‌توانید از کتابخانه‌های موجود استفاده کنید. در صورت استفاده از هر کتابخانه‌ای، باید پارامترهای مربوطه و دلیل مقدار دهی هر پارامتر را شرح دهید.

در بخش GMM، توابع زیر پیاده سازی شده است:

- get_LPC_GMM_model_of_speaker: تابعی که دیتافریم آموزشی مربوط به یک گوینده را گرفته و بر اساس ویژگی‌های lpc آن مدل می‌سازد. برای ۳۲ مولفه‌ای بودن باید پارامتر n_component (تعداد گوسین) آن را برابر با ۳۲ و برای قطری بودن از covariance_type = 'diag' استفاده میکنیم. همچنین آموزش آن در ۱۰۰ دور انجام میشود.
- get_MFCC_GMM_model_of_speaker: تابعی که دیتافریم آموزشی مربوط به یک گوینده را گرفته و بر اساس ویژگی‌های MFCC آن مدل می‌سازد. برای ۳۲ مولفه‌ای بودن باید پارامتر n_component (تعداد گوسین) آن را برابر با ۳۲ و برای قطری بودن از covariance_type = 'diag' استفاده میکنیم. همچنین آموزش آن در ۱۰۰ دور انجام میشود.

در دو تابع فوق فریم‌ها همگی با هم concatenate میشوند.

- Get_GMMs: مدل‌های GMM برای هر یک از ویژگی‌های LPC و MFCC برای تمامی گویندگان (۲۰ مدل برای هر ویژگی که می‌شود ۴۰ مدل) بر می‌گرداند.

در این بخش مدل‌ها ساخته شده و در قالب لیست‌هایی نگه داشته شده است.

۵. در نهایت به ازای هر فایل صوتی در مجموعه داده تست، برچسب گوینده آن را با استفاده از دو مدل ایجاد شده، مشخص کنید. نتیجه باید در دو حالت Top-1 و Top-3 به ازای هر مدل مشخص شود.
   a. Top-1 به معنای آن است که اگر اولین برچسب برای یک فایل (برچسب صوتی متناظر با بالاترین احتمال) همان برچسب واقعی گوینده باشد، یک شناسایی درست حساب می‌شود.
   b. Top-3 به معنای آن است که اگر هرکدام از ۳ بچسب اولیه (متناظر با سه مقدار بیشترین احتمال) برچسب واقعی گوینده بود، یک شناسایی درست حساب می‌شود.

در بخش predict speakers، تابع Predict و Predict_top_3 پیاده‌سازی شده است که در آن با توجه به score داده شده برای هر مدل و برای هر گوینده‌ی آزمایشی، مقادیر sort شده و سه تای بهترین برگردانده شده است.سپس برای هر یک از سطرهای test_df پیش‌بینی انجام شده است، سپس دیتافریمی تحت عنوان pred_df که شامل ستون‌های real و lpc_pred و mfcc_pred برای تمامی سطرهای test_df ساخته شده است.

۶. دو مدل ایجاد شده را از نظر دقت با یکدیگر مقایسه کنید.

در بخش predict speakers، تابع accuracy نوشته شده است که در آن دقت GMM برای ویژگی و top_k ورودی محاسبه می‌شود. نتایج بدست آمده برای هر کدام از حالات ذکر شده به صورت زیر است:

```
accuracy of top_1 for GMM with lpc featuers= 0.85
accuracy of top_3 for GMM with lpc featuers= 0.9166666666666666
accuracy of top_1 for GMM with mfcc featuers= 0.9333333333333333
accuracy of top_3 for GMM with mfcc featuers= 0.9666666666666667
```

همانطور که نشان داده شده است ویژگی‌های MFCC از ویژگی‌های LPC برای تشخیص گوینده کارآمدتر بوده است.

همچنین اگر مدل نتوانسته در گزینه اول تشخیص دهد به احتمال خیلی زیاد در سه گزینه‌ی اول انتخابی‌اش توانسته درست پیش‌بینی کند.

۷. علاوه بر گزارش نهایی و کدهای پروژه، باید فایل‌های زیر را نیز ارسال کنید:

a. ماتریس ضرایب MFCC نرمال شده و مشتقات اول و دوم در قالب mat. یا joblib. با نام MFCC برای فایل هر گوینده در دو مجموعه یادگیری و تست.

b. ماتریس ویژگی‌های LPC در قالب mat. یا joblib. با نام LPC برای فایل هر گوینده در دو مجموعه داده یادگیری و تست.

c. مدل GMM ذخیره شده برای هر گوینده بر مبنای ویژگی‌های MFCC و همینطور LPC.

در بخش save matrixes ماتریس ویژگی‌های MFCC، LPC برای train و test ذخیره شده است. این فایل‌ها با نام مناسب در پوشه‌ی outs آمده است. همچنین برای اطمینان از درست ذخیره شدن هر کدام یک بار load شده و نشان داده شده است.

در بخش save models نیز هر مدل برای هر گوینده و با هر یک از ویژگی‌های MFCC و LPC با استفاده از pickle ذخیره شده است. این فایل‌ها با نام مناسب در پوشه‌ی outs آمده است. همچنین برای اطمینان از درست ذخیره شدن هر کدام یک بار load شده و نشان داده شده است.