Technische Universität Berlin

Grundlagen der automatischen Spracherkennung

Aufgabe 4 – MFCCs

22.11.2023

Wentao Yu

Technische Universität Berlin

Planung

- **Heute**: Extraktion von MFCC-Merkmalen, siehe Übungsblatt. Die Signalverarbeitung wird damit abgeschlossen.

- In der kommenden Woche können Sie in der Übung Ihre Merkmalsextraktion fertigstellen.

- **Vom 27 Nov. um 0 Uhr bis zum 30 Nov. um 24 Uhr** haben Sie dann die Möglichkeit, freiwillig Ihren Code zur Merkmalsextraktion **IN DER GRUPPE** abzugeben. Es erfolgt **KEINE** Bewertung; stattdessen erhalten Sie von uns Kommentare.

Technische Universität Berlin

MFCCs

1. compute_cepstrum(mel_spectrum, num_ceps)

2. get_delta(x)

3. append_delta(x, delta)

Erweitern Sie schließlich die Funktion compute_features().

Hinweise zur Implementierung, finden Sie im Übungsblatt Aufgabe4.pdf im ISIS-Kurs.

Technische Universität Berlin

Benötigte Funktionen in feature_extraction.py

```
def compute_cepstrum(mel_spectrum, num_ceps):
    ...
```

$$x_{CEP}[\tau] = \mathrm{DCT}(\log\{|x_{MEL}[\tau]|\})$$

Diese Funktion berechnet das reele Cepstrum aus einem gegebenen Mel-Spektrum. Um numerische Probleme zu vermeiden, sollten Nullen im Mel-Spektrum durch den kleinsten darstellbaren Wert ersetzt werden.

- mel_spectrum: $\boldsymbol{x}_{MEL}$, wird von Funktion apply_mel_filters() zurückgegeben.

- num_ceps: Anzahl der zurückzugebenden Koeffizienten.

Technische Universität Berlin

Benötigte Funktionen in feature_extraction.py

```
def get_delta(x):
    ...
```

Diese Funktion berechnet die erste zeitliche Ableitung von einem Merkmalsvektor, zum Beispiel vom Cepstrum $\boldsymbol{x}_{CEP}$.

$$\Delta \boldsymbol{x}_{\mathrm{CEP}}[\tau] = \frac{1}{2}\Big(\boldsymbol{x}_{\mathrm{CEP}}[\tau+1] - \boldsymbol{x}_{\mathrm{CEP}}[\tau-1]\Big)$$

$$\Delta \boldsymbol{x}_{\mathrm{CEP}}[0] = \boldsymbol{x}_{\mathrm{CEP}}[1] - \boldsymbol{x}_{\mathrm{CEP}}[0]$$

$$\Delta \boldsymbol{x}_{\mathrm{CEP}}[T-1] = \boldsymbol{x}_{\mathrm{CEP}}[T-1] - \boldsymbol{x}_{\mathrm{CEP}}[T-2]$$

Technische Universität Berlin

Benötigte Funktionen in feature_extraction.py

```
def append_delta(x, delta):
    ...
```

Diese Funktion konkateniert einen Merkmalsvektor x mit dessen erster Ableitung delta.

Technische Universität Berlin

Erweitern von compute_features()

```
def compute_features(audio_file, window_size=25e-3, hop_size=10e-3,
                     feature_type='STFT', n_filters=24, fbank_fmin=0,
                     fbank_fmax=8000, num_ceps=13):
    ...
```

Erweitern Sie die compute_features() Funktion:

- Wenn feature_type gleich MFCC, sollten die MFCCs zurückgegeben werden.
- Wenn feature_type gleich MFCC_D, sollten MFCCs und deren erste zeitliche Ableitung konkateniert & zurückgegeben werden.
- Wenn feature_type gleich MFCC_D_DD, sollten MFCCs, erste und zweite zeitliche Ableitung konkateniert und als Ausgabe zurückgegeben werden.

Technische Universität Berlin

Erstellen des Hauptskripts: uebung4.py

Verwenden Sie die compute_features() Funktion, um die MFCCs sowie deren erste und zweite zeitliche Ableitung für die Audiodatei 'TEST-MAN-AH-3O33951A.wav' zu berechnen. Plotten Sie die extrahierten MFCC-Merkmale unter Verwendung der Standardeinstellungen der Funktion compute_features().

Technische Universität Berlin

Abgabe

In der nächsten Woche können Sie Ihren Code in einer ZIP-Datei in ISIS hochladen.

Die Abgabe sollte die folgende Struktur haben:

./recognizer/__init__.py

./recognizer/feature-extraction.py

./recognizer/tools.py

./uebung1.py

./uebung2-3.py

./uebung4.py

Technische Universität Berlin

Melden Sie sich gerne bei Fragen