Technische Universität Berlin

Grundlagen der automatischen Spracherkennung

Aufgabe 1 – Merkmalsextraktion I

01.11.2023

Wentao YU

Technische Universität Berlin

ISIS-Kursdatei in Git-Repositories aktualisieren

Im ISIS-Kurs ist das Übungsblatt Aufgabe1.pdf mit den genauen Aufgabenstellungen.

Außerdem ist dort die Datei **uebung1.zip**.

Schritte :

1. cd ./my-repository
2. git pull
3. Entpacken Sie die ZIP-Datei in die lokale Kopie Ihres Repositories mit der folgenden Struktur:

   ./my-repository/data/TEST-MAN-AH-3033951A.wav

   ./my-repository/uebung1.py

   ./my-repository/recognizer/__init__.py

   ./my-repository/recognizer/feature-extraction.py

   ./my-repository/recognizer/tools.py

Technische Universität Berlin

Dateistruktur des Spracherkenners (erster Teil)

--recognizer

feature_extraction.py

tools.py

Technische Universität Berlin

Benötigte Funktionen in tools.py

```
def sec_to_samples(x, sampling_rate):
    ...
```

Diese Funktion berechnet, wie viele Samples es in einem Zeitintervall gibt.

x: Zeitintervall in Sekunden

sampling_rate: Abtastfrequenz

Technische Universität Berlin

Benötigte Funktionen in tools.py

```
def next_pow2(x):
    ...
```

p = next_pow2(A) gibt die kleinste Zweierpotenz zurück, sodass 2^p **größer oder gleich** dem Absolutwert von A ist.

next_pow2(300) = 9

$2^8 = 256 < 300$

$\mathbf{2^9 = 512 > 300}$

Technische Universität Berlin

Benötigte Funktionen in tools.py

```
def dft_window_size(x, sampling_rate):
    ...
```

Berechnen Sie die nächste Zweierpotenz der Anzahl der Samples für eine gegebene Länge in Sekunden und eine gegebene Abtastfrequenz.

Z.B. sollte für x = 1 und sampling_rate = 1000 als Ergebnis 1024 zurückgegeben werden.

Benötigte Funktionen in tools.py

```
def get_num_frames(signal_length_samples, window_size_samples,
                   hop_size_samples):
    ...
```

$$K = \left\lceil \frac{Q - O}{R} \right\rceil, \qquad \lceil \quad \rceil$$: Aufrundung

K: Anzahl der Rahmen
Q: Signallänge -> signal_length_samples
N: Fensterlänge -> window_size_samples
R: Rahmenvorschub -> hop_size_samples
O: Overlap = *N* - *R*

Technische Universität Berlin

Benötigte Funktionen in feature_extraction.py

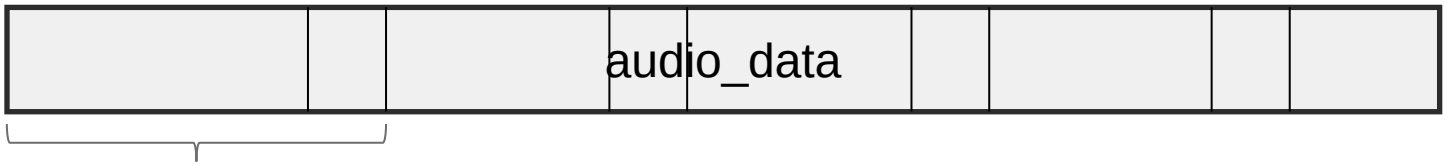
```
def make_frames(audio_data, sampling_rate, window_size, hop_size):
    ...
```

1. Segmentieren Sie das Audiosignal mit gleicher Fensterlänge und gleicher Überlappungslänge.
2. Die Fensterlänge N soll hierbei zunächst so modifiziert werden, dass sie im zeitdiskreten Bereich der nächsthöheren Zweierpotenz entspricht (dft_window_size(window_size, sampling_rate)).
3. Multiplizieren Sie die Rahmen mit einem Hamming-Fenster der Länge N.
4. Für den letzten Rahmen wird evtl. Zero-Padding benötigt

Benötigte Funktionen in feature_extraction.py

```
def make_frames(audio_data, sampling_rate, window_size, hop_size):
    ...
```

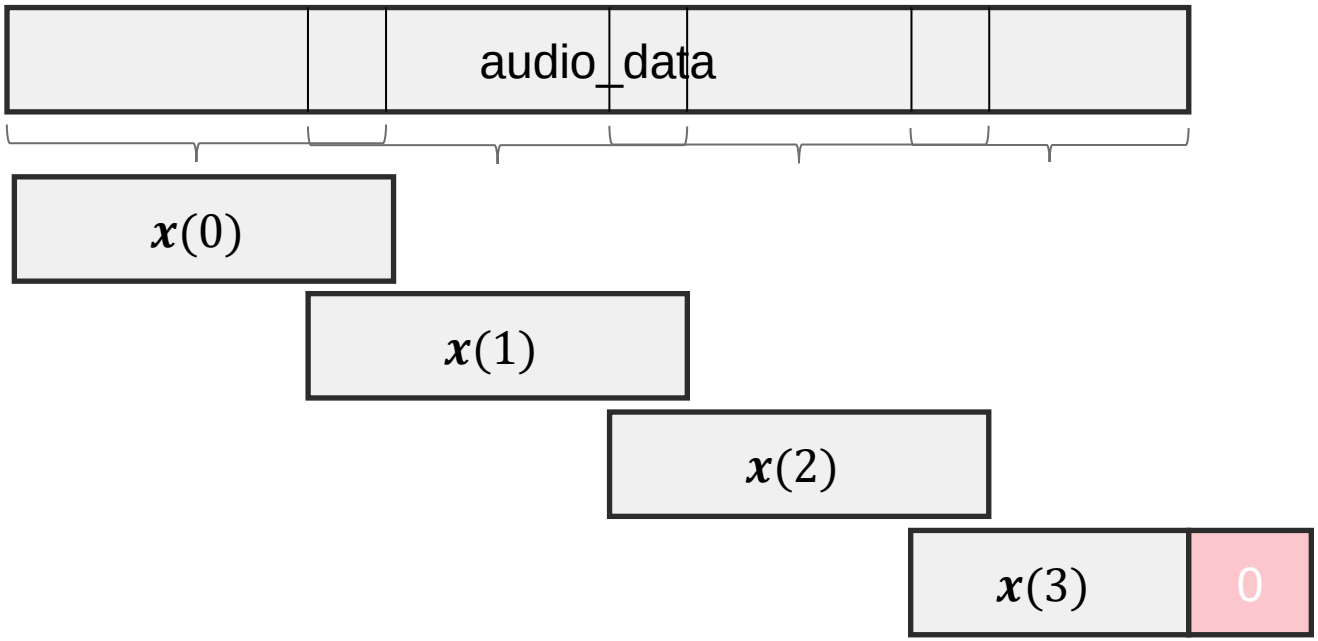
R = hop_size



N = window_size

Benötigte Funktionen in feature_extraction.py

```
def make_frames(audio_data, sampling_rate, window_size, hop_size):
    ...
```



Benötigte Funktionen in feature_extraction.py

```
def make_frames(audio_data, sampling_rate, window_size, hop_size):
    ...
```

Hamming = numpy.hamming(window_size)

Technische Universität Berlin

Benötigte Funktionen in feature_extraction.py

```
def make_frames(audio_data, sampling_rate, window_size, hop_size):
    ...
```

Rückgabe: zweidimensionales Float-Array $\boldsymbol{x}_w \in \mathbb{R}^{k \times d}$

k: Anzahl der Rahmen

d: modifizierten Fensterlänge

Technische Universität Berlin

Erstellen der Hauptfunktion: uebung1.py

1. Lesen Sie die Audiodatei 'TEST-MAN-AH-3O33951A.wav' in Python mithilfe der Python-Funktion scipy.io.wavfile.read() ein

2. Wenden Sie die Funktion make_frames() auf das eingelesene Audiosignal an.

3. Plotten Sie die segmentierten Rahmen in Subplots mithilfe der Python Matplotlib-Bibliothek. Das ist der kreative Teil für heute ;)

Ein Beispiel wie ein Ergebnis aussehen könnte, und mehr Hinweise zur Implementierung, finden Sie im Übungsblatt Aufgabe1.pdf im ISIS-Kurs.

Technische Universität Berlin

Git-Repositories aktualisieren

1. git add .
2. git commit –m ‘Aufgabe 1‘
3. git push

Technische Universität Berlin

Melden Sie sich gerne bei Fragen.