Technische Universität Berlin

Grundlagen der automatischen Spracherkennung

Aufgabe 6 – DNN Training I: Vorbereitung der Daten

20.12.2023

Wentao Yu

Technische Universität Berlin

Vorbereitung

TIDIGTS-ASE Datensatz herunternladen:
https://tubcloud.tu-berlin.de/s/y3RDt5x9JGmmrRB

Aufgabe6.zip aus dem ISIS-Kurs:

./Aufgabe6/dataset/train.json
./Aufgabe6/dataset/dev.json
./Aufgabe6/dataset/test.json
./Aufgabe6/recognizer/hmm.py
./Aufgabe6/recognizer/tools.py

Technische Universität Berlin

Vorbereitung

Kopieren Sie:

- ./Aufgabe6/dataset-Ordner in Ihr lokales ./my-repository/
- Alle vier Funktionen in ./Aufgabe6/recognizer/tools.py in Ihre eigene recognizier/tools.py-Datei
- ./Aufgabe6/recognizer/hmm.py in Ihr eigenes recognizer-Verzeichnis

Technische Universität Berlin

Benötigte Funktionen in feature_extraction.py

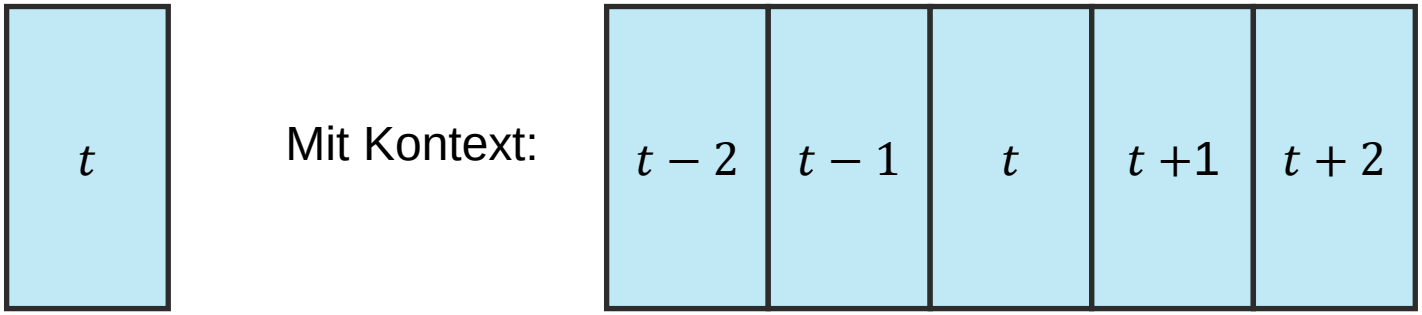
```
def add_context(feats, left_context=6, right_context=6):
    ...
```

Diese Funktion fügt Vorgänger- und Nachfolger-Kontext hin.

- Die Eingabe feats: [f_len, f_dim], wobei f_len die Rahmensequenzlänge, f_dim die Merkmalelänge.
- Die Ausgabe: [f_len, f_dim, c_dim], wobei c_dim = left_context + right_context +1.
- Für den ersten und den letzten Rahmen soll mit dem Kopieren des ersten bzw. letzten Rahmens aufgefüllt werden
- left_context und right_context können auch Null sein.

Benötigte Funktionen in feature_extraction.py

Left_context=2, right_context=2



[1, f_dim]

[1, f_dim, c_dim]=[1, f_dim, 5]

Technische Universität Berlin

Benötigte Funktionen in feature_extraction.py

```
def compute_features_with_context(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,
                                  left_context=4, right_context=4):
    ...
```

Diese Funktion führt die Merkmalsextraktion durch und dann entsprechend den Kontext anhängt und die extrahierten Features mit Kontext zurückgibt.

Technische Universität Berlin

Data-Loader konstruieren

- Verwenden Sie die Aufgabe 5, utils.py, als Beispiel und erstellen Sie ein neues Skript: ./my-repository/recognizer/utils.py

- Extrahieren Sie die Features mit der Funktion compute_features_with_context()

- Extrahieren Sie die Ground-Truth-Labels mit der Funktion tools.praat_file_to_target()

- Für eine genauere Beschreibung lesen Sie bitte das Übungsblatt.

Technische Universität Berlin

Erstellen des Hauptskripts: uebung6.py

- Erstellen Sie einen Data-Loader für das Development-Set.
- Iterieren Sie durch Ihren Data-Loader und plotten Sie die Ground-Truth-Labels der ersten beiden Samples, die vom Data-Loader zurückgegeben wurden
- Überprüfen Sie, ob die Plotten unterschiedlich sind.

Ein Beispiel wie ein Ergebnis aussehen könnte, und mehr Hinweise zur Implementierung, finden Sie im Übungsblatt Aufgabe6.pdf im ISIS-Kurs.

Technische Universität Berlin

Repository-Struktur

./my-repository/data/TEST-MAN-AH-3033951A.wav
./my-repository/dataset/train.json
./my-repository/dataset/dev.json
./my-repository/dataset/test.json
./my-repository/recognizer/__init__.py
./my-repository/recognizer/hmm.py
./my-repository/recognizer /tools.py
./my-repository/recognizer/feature-extraction.py
./my-repository/recognizer/utils.py

./my-repository/torch_intro/dataset/train.json
./my-repository/torch_intro/dataset/dev.json
./my-repository/torch_intro/dataset/test.json
./my-repository/torch_intro/local/feature-extraction.py
./my-repository/torch_intro/local/model.py
./my-repository/torch_intro/local/utils.py
./my-repository/torch_intro/local/train.py

./my-repository/uebung1.py
./my-repository/uebung2.py
./my-repository/uebung3.py
./my-repository/uebung4.py
./my-repository/uebung5.py
./my-repository/uebung6.py

Technische Universität Berlin

Melden Sie sich gerne bei Fragen