



Grundlagen der automatischen Spracherkennung

Aufgabe 7 – DNN Training II

10.01.2024

Wentao Yu



Hauptskript uebung7.py erstellen

Erstellen Sie auf Basis Ihrer Datei uebung5.py ein neues Hauptskript uebung7.py in Ihrem Repository.

- Gute Parameter für die Merkmalsextraktion finden Sie in Aufgabe 6.4
- Batchgröße: Ab jetzt immer 1

Modell entwerfen

```
class DNN_Model(torch.nn.Module):  
    def __init__(self, idim, odim, hidden_dim):  
        ...  
    def forward(self, audio_feat):  
        ...
```

Verwenden Sie Ihre Datei `model.py` aus Aufgabe 5 als Beispiel und erstellen Sie ein neues Skript: `./my-repository/recognizer/model.py`

Modell:

- Eingabe-Dimension: `[BS, f_len, f_dim, c_dim]`
- Flatten-Layer
- Feedforward-Netzwerk
- Klassifizierungsschicht



Training und Evaluierung eines Klassifikators

Verwenden Sie Ihre Datei `train.py` aus Aufgabe 5 als Beispiel und erstellen Sie ein neues Skript: `./my-repository/recognizer/train.py`

- Kreuzentropie als Loss-Funktion
- Batchgröße auf 1 festgelegt.
- Es ist nicht erforderlich, Zwischenergebnisse in JSON zu speichern.

Implementieren Sie die `train()` und `evaluation()` Funktionen, um das Modell zu trainieren und evaluieren.

Für eine genauere Beschreibung lesen Sie bitte das Übungsblatt.



Benötigte Funktionen in train.py

```
def wav_to_posteriors(model, audio_file, parameters):  
    ...
```

Diese Funktion soll die geschätzten a-posteriori Wahrscheinlichkeiten eines Audio-Files über den Frames und HMM-Zuständen berechnen.

Plotten Sie am Ende der `run()` Funktion Ihrer Datei `train.py` diese Posteriors für die Beispieldatei `TEST-WOMAN-BF-7O17O49A.wav` zusammen mit den Ground-Truth-Labels (die Sie mit der Funktion `praat_file_to_target()` erhalten).

Ein Beispiel wie ein Ergebnis aussehen könnte, und mehr Hinweise zur Implementierung, finden Sie im Übungsblatt `Aufgabe7.pdf` im ISIS-Kurs.



Melden Sie sich gerne bei Fragen