HANSER

Erratum zu

KI-Sprachassistenten mit Python entwickeln

von Jonas Freiknecht

Print-ISBN: 978-3-446-47231-0
E-Book-ISBN: 978-3-446-47448-2
E-Pub-ISBN: 978-3-446-47659-2

TIPP: Wenn Sie an einem Punkt in diesem Kapitel angelangt sind, an denen Ihnen das Training zu aufwendig ist oder die nötige Zeit oder Hardware fehlt, dann laden Sie einfach ein von mir fertig trainiertes Modell herunter. Sie finden diese in den Releases des von uns genutzten Repositorys:

https://github.com/padmalcom/Real-Time-Voice-Cloning-German/releases

Im nächsten Abschnitt klonen wir Repositorys. Wenn Sie ein Release runtergeladen haben, kopieren Sie einfach die gesamte Struktur in das Repository, sodass Sie die Dateien *encoder/saved_models/my_run.pt*, *synthesizer/saved_models/my_run/my_run.pt* und *vocoder/saved_models/my_run/my_run.pt* vorliegen haben. Ist das erledigt, können Sie zu Abschnitt 3.3.9 springen.

Glücklicherweise gibt es für das Training von sprachrelevanten Aufgaben das Speech Dataset von M-AILABS, in dem uns Forscher und IT-Spezialisten etwa 20 GB Sprachdaten samt textueller Transkription zu Verfügung stellen, die wir fast ohne Anpassung für unseren Zweck verwenden können. Während Sie nun weiterlesen, können sie bereits hier den Download anstoßen, denn der wird eine Weile benötigen:

https://github.com/imdatceleste/m-ailabs-dataset

Suchen Sie nach der Überschrift *Statistics & Download Links* und klicken Sie neben der Sprache *GERMAN* ganz rechts auf *DOWNLOAD*.

Neben der Herausforderung der Datenlage sehen wir uns noch mit einer weiteren konfron-tiert. Wir werden relativ lange trainieren müssen, bis wir die Güte eines Modells bestimmen, und noch länger, um die Qualität aller Modelle im Zusammenspiel bewerten zu können. So kann es sein, dass der Encoder nach vier bis fünf Tagen anzeigt, dass das Training keine weiteren Verbesserungen mit sich bringt, ob aber das Embedding letztendlich Einfluss auf die ausgegebene Stimme nehmen kann, erfahren Sie erst nach etwa zehn weiteren Tagen nach dem Training des Synthesizers und des Vocoders. Ein Umstand, der im Kontext von Deep Learning häufig für Frustration sorgt und dem nur mit ausreichender Planung und Konzeption zu begegnen ist. Wenn Sie (wie ich) eher der experimentelle Mensch sind, der Dinge ausprobiert und im Falle des Scheiterns einen anderen Weg geht, bis etwas funktio-niert, dann erfordert es hier sicherlich ein Umdenken. Aber derartige Herausforderungen halten uns mental fit, nicht wahr?

Zum Vorgehen finden Sie nachfolgend ein kurze Liste:

- Einrichten des Projekts
- Beschaffen der Trainingsdaten
- Formatieren der Trainingsdaten
- (Optional) Training des Encoders
- Training des Synthesizers
- (Optional) Training des Vocoders
- Anwendung unserer Modelle über die UI und per Code

Wir werden also zunächst das Projekt einrichten und die Trainingsdaten entpacken und aufbereiten. Dann beginnen wir mit dem Training des Encoders, das optional ist für den Fall, dass Sie keine Stimmen imitieren, sondern die des trainierten Modells verwenden möchten.