Building a Real Time Voice Transfer App with Streamlit and Python

dataroots

Real Time Voice Transfer

dataroots

What is Voice Transfer?

- Voice cloning
- Artificial simulation of a person’s voice
- Applications
  - For people who lost their voice
  - Transferring a voice across languages
  - Generate speech from text in low resource settings

dataroots

Context: Voice Cloning

- Large amounts of high quality recordings is **impractical for many speakers**
- Deep neural network trained on a corpus of hours of recorded speech from a single speaker
- **Giving a new voice to such a model**
  - highly expensive
  - record a new dataset
  - retrain the model

Goal: Transfer Learning

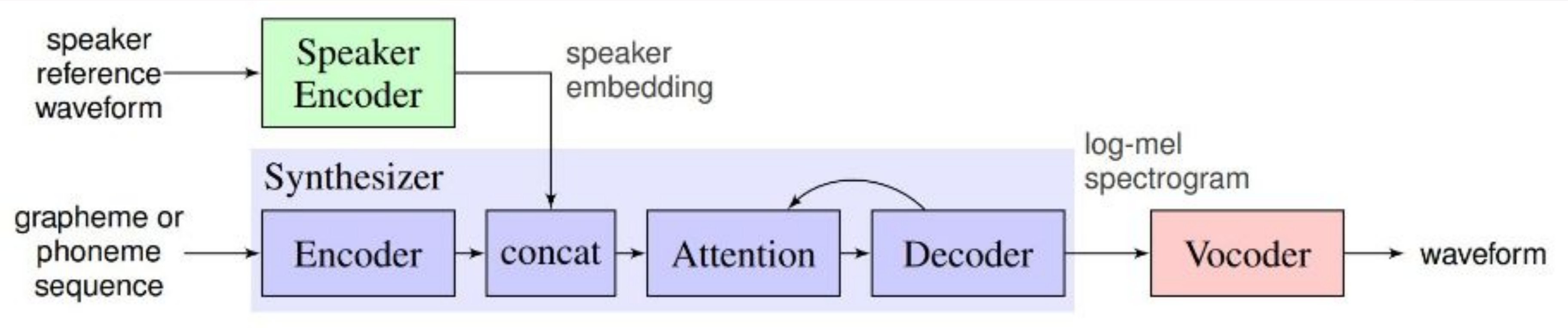
- Key idea of Transfer Learning
  - transfer knowledge from one task with a lot of labelled data
  - to related tasks with very little labelled data
- Text to speech
- Zero-shot setting
  - transfer to voices unseen in the training set

Approach: Voice Cloning

- Decouple speaker modeling from speech synthesis
- Speaker-discriminative embedding network
- Text to speech network
    - conditioned on embedding unique to speaker

Framework: Overview

- Real-time Voice Cloning (Jia et al., 2018)
  - A speaker encoder: GE2E loss (Wan et al., 2017)
  - A synthesizer: Tacotron (Wang et al., 2017)
  - A vocoder: Wavenet (van den Oord et al., 2016)

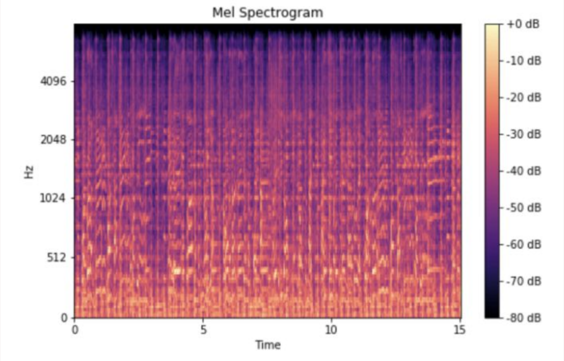


dataroots

Figure from Jemine (2019).

Stage 1: Speaker Encoder

- A speaker encoder: GE2E loss (Wan et al., 2017)
    - The reference speech is a sequence of log-mel spectrogram from a speech utterance
    - Embedding captures the unique characteristics of the speaker
    - Embeddings of utterances from the same speaker have high cosine similarity



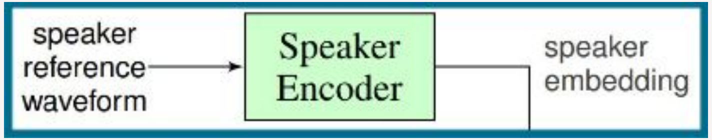
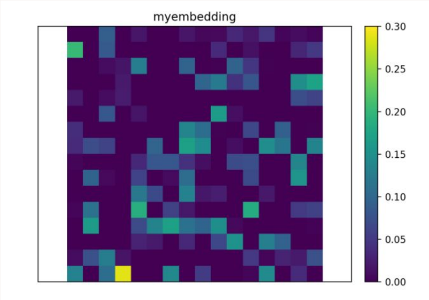
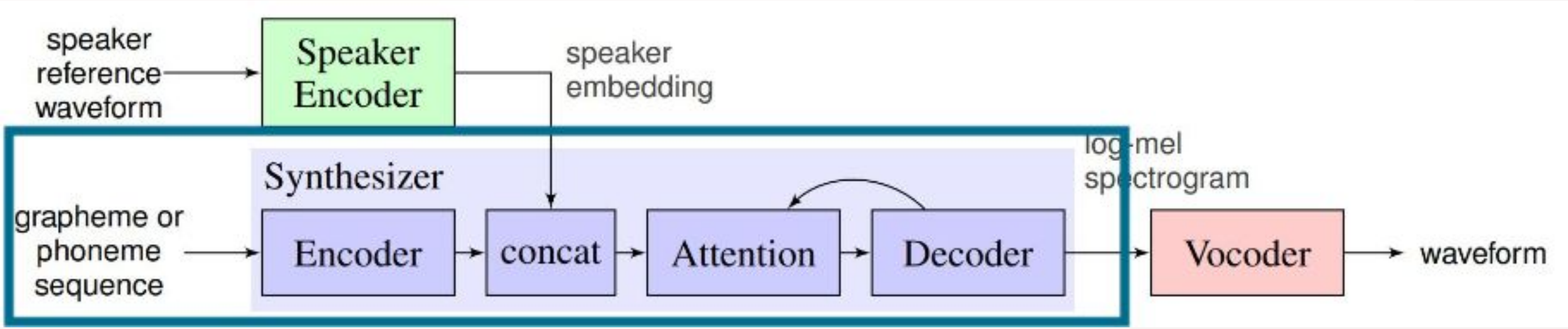


Figure from Jemine (2019).



Stage 2: Synthesizer

- Synthesizer: Tacotron (Wang et al., 2017)
  - Extend attention Tacotron 2 to support multiple speakers (Jia et al., 2018)
  - Embedding vector is concatenated with the synthesizer encoder output at each time step.

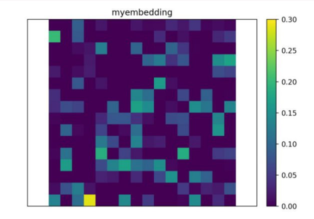


datar∞ts

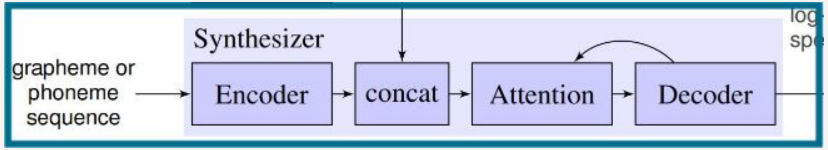
Figure from Jemine (2019).

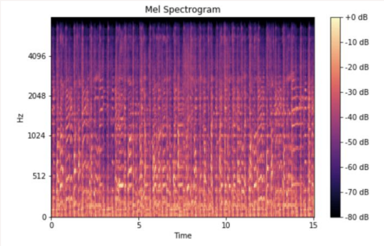
Stage 2: Synthesizer

- Synthesizer: Tacotron (Wang et al., 2017)
  - The synthesizer is trained on pairs of text transcript and target audio.
  - The text is mapped to a sequence of phonemes,
  - Trained in a transfer learning configuration



"Hello world"





dataroots

Stage 2: Synthesizer

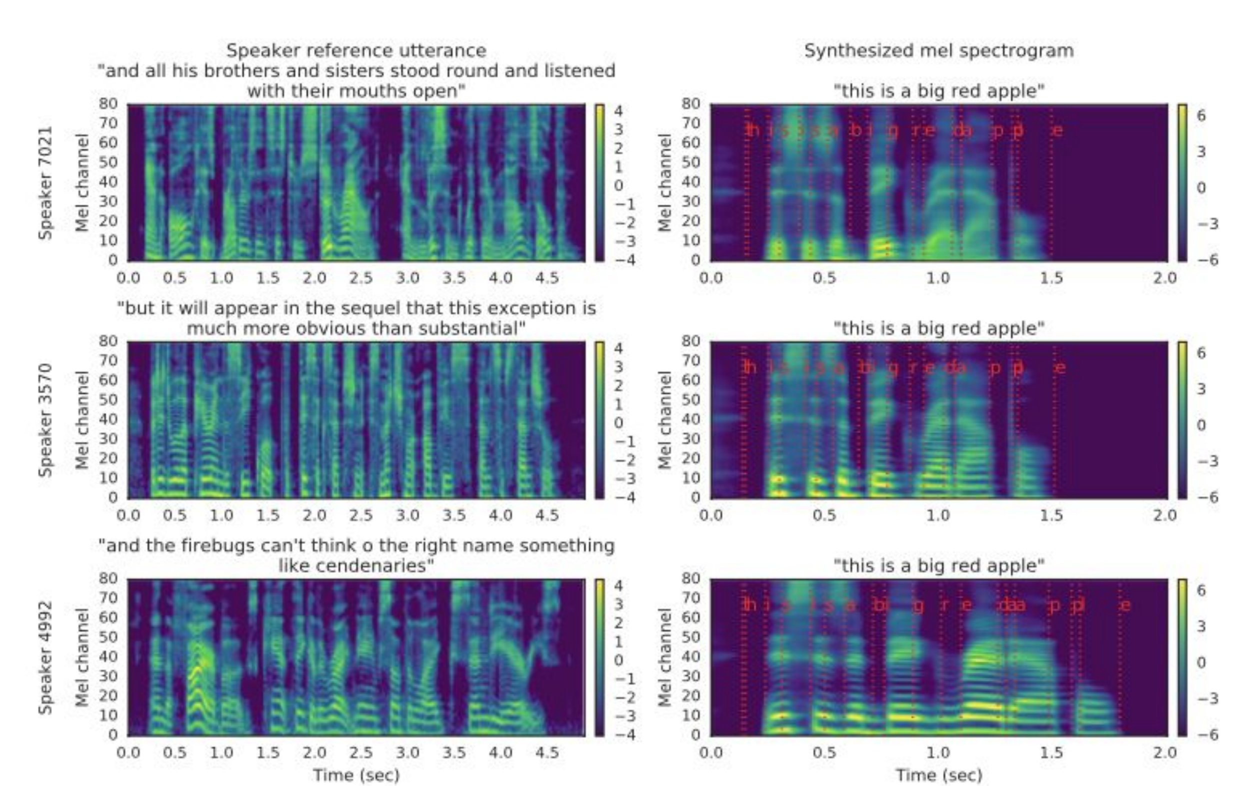
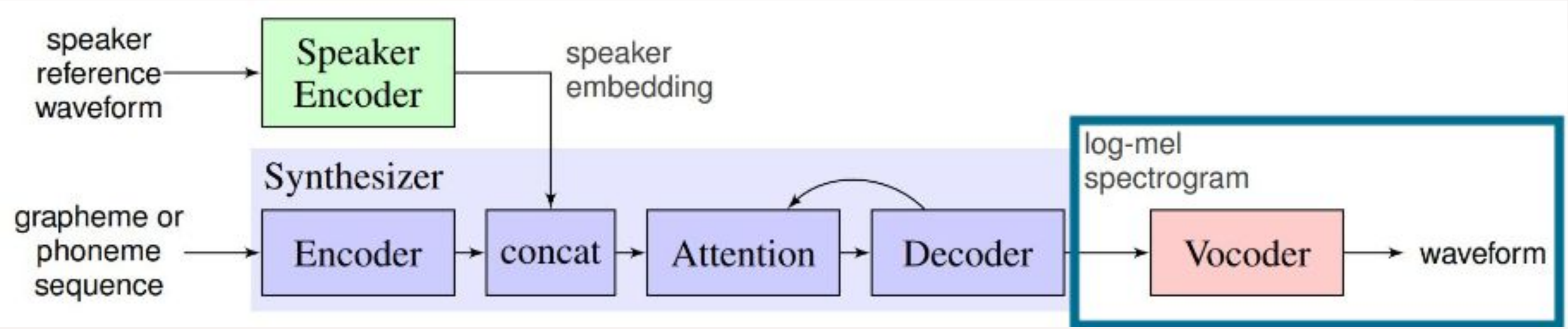


Figure from Jia et al. (2018)

datarootsts

Stage 3: Vocoder

- Vocoder WaveNet (Kalchbrenner et al., 2018)
  - autoregressive WaveNet [19] as a vocoder to invert synthesized mel spectrograms into time-domain waveforms

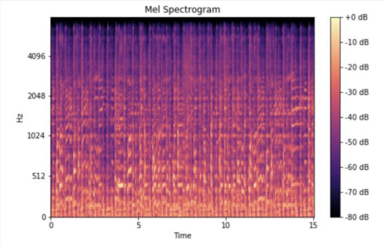


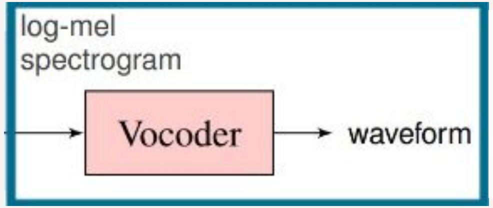
dataroots

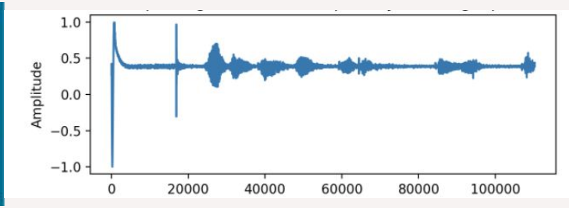
Figure from Jemine (2019).

Stage 3: Vocoder

- Vocoder WaveNet (Kalchbrenner et al., 2018)
  - synthesized mel spectrogram captures all details for high quality synthesis of a variety of voices
  - allowing a multispeaker vocoder by training on data from many speakers







dataroots

Datasets

- VCTK
  - 44 hours of clean speech
  - 109 speakers
  - British accents
- LibriSpeech
  - 2 clean training sets
  - comprising 436 hours of speech
  - from 1,172 speakers

Building Voice Transfer with Streamlit

dataroots

What is Streamlit?

- Data scientists build apps
  - dashboard, data browser, etc.
- Ad hoc building flow
  - jupyter notebook > python script > flask app > need more features...
  - maintainability

What is Streamlit?

- Streamlit is an app framework for data scientists
- Key Idea
  - Make webapps as easy as writing python scripts
  - Use traditional iterative scripting process
  - Instead of layout and event flow
- Workflow
  - Start with python script
  - Slightly annotate to make it an app

dataroots

What is Streamlit?

- Embrace python scripting
  - everything you can do in a python script
  - you can do in streamlit
- Treat widgets as variables
  - substitute variables with a widget such as st.slider()
  - reuse variables as widgets iteratively
- Reuse data and computation
  - cache computation

dataroots

Demo

More info

- https://www.streamlit.io/
- https://github.com/datarootsio/rootslab-streamlit-demo

References (1)

- Jia, Ye, et al. "Transfer learning from speaker verification to multispeaker text-to-speech synthesis." Advances in neural information processing systems. 2018.
- Wan, Li, et al. "Generalized end-to-end loss for speaker verification." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

References (2)

- Shen, Jonathan, et al. "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.
- Kalchbrenner, Nal, et al. "Efficient neural audio synthesis." arXiv preprint arXiv:1802.08435 (2018).

References (3)

- Jemine, C. (2019). Master thesis : Real-Time Voice Cloning. (Unpublished master's thesis). Université de Liège, Liège, Belgique. Retrieved from https://matheo.uliege.be/handle/2268.2/6801

dataroots

The End

dataroots