**SPEECH SYNTHESIS USING DEEP LEARNING**

1. **Abstract**

The artificial production of human speech is believed as speech synthesis. Text-to-speech, music generation, voice generation, speech-enabled devices, navigation systems, and accessibility for visually impaired people can all profit from in this machine learning-based technique.

Deep Learning may be a collection of machine learning that consists of algorithms galvanized by the structure and performance of the human brain said as neural networks. Deep Learning has been same to make revolutionary advances inside the machine learning and AI field.

In this project, we aim to form a speech synthesizer victimization Deep Neural Networks. The program would extract some characteristics from several samples of the speaker's voice, clone it and generate audio supported the given text. The performance of the program is about by the naturalness of speech and conjointly the similarity between the speaker’s speech and conjointly the recreated speech.

1. **Introduction**

Deep learning is a subset of Machine Learning which aims to imitate the functioning of the human brain and learn from large amounts of data. Then, the same task is repeatedly performed so that the model learns from experience and improves its performance in subsequent attempts.

Deep learning is very popular in many sub-disciplines of machine learning. It is predominant in text-to-speech, which is the assistive technology that converts text to spoken word. A lot of research has been done to make these deep learning models more efficient and generate more natural-sounding speech. Some models even produce artificial speech that is almost indiscernible from human speech. Interestingly, since speech naturalness is judged based on subjective metrics, artificial speech that is “more natural than human speech” might be possible to produce. In fact, some even argue that it has already been achieved.

Speech generation and text-to-speech have a wide variety of applications and become increasingly popular every day. Properly and professionally recorded speech with correct pronunciation and sufficient variations in pitch would help the model produce more accurate speech with minimal background noise. However, not many datasets with properly recorded speech for models to use as input data exist currently. Training a relatively basic text-to-speech model such as Tacotron would require hundreds of hours of sample input speech.

Voice conversion is the process of converting a speech segment from one voice to another. On the other hand, voice cloning is the process of capturing a speaker’s voice and processing speed. To determine the similarity of generated voice, input reference samples needed can range from half an hour-long to only a few seconds long.

This study aims to work towards an efficient and potent voice cloning model. It should be able to clone voices with only a few seconds of reference input audio. we will apply the work of (J. Corentin et al, 2018) along with my own implementation and dataset.

1. **Related work**

Deep models that produce natural-sounding speech began appearing in 2016. Since then, a lot of research has been done in the field to make them better across many characteristics like efficiency, naturalness, etc.

Some of these include:

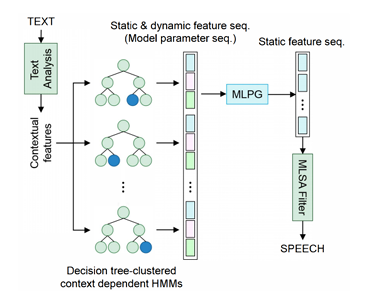
* Sercan Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech, 2017.
* J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In INTERSPEECH, 2018.

This study was based on the following paper:

* J. Corentin: Automatic Multispeaker Voice Cloning, 2018.

1. **Research Methods**

Hidden Markov Model (HMM) based framework approach, consists of clustering the linguistic features extracted from the input text with a decision tree, and training an HMM per cluster. The HMMs are tasked to produce a distribution over spectrogram coefficients, their derivative, second derivative, and a binary flag that indicates which parts of the generated audio should contain voice. With the Maximum Likelihood Parameter Generation algorithm (MLPG) spectrogram coefficients are sampled from this distribution and eventually fed to the MLPG vocoder. It is possible to modify the voice generated by conditioning the HMMs on a speaker or tuning the generated speech parameters with adaptation or interpolation techniques.



*Fig: HMM-based TTS pipeline*

Improvements to this framework were later brought by feed-forward Deep Neural Networks (DNN-2013) which has better data efficiency as the training set is no longer fragmented in different clusters of contexts.

RNNs(2014) make natural acoustic models as they are able to learn a compact representation of complex and long-span functions. As RNNs (Recurrent Neural Network) are fit to generate temporally consistent series, the static features can directly be determined by the acoustic model, alleviating the need for dynamic features and MLPG.

WaveNet(2016) is a deep convolutional neural network that, for a raw audio waveform, models the distribution of a single sample conditionally to previous ones made a substantial breakthrough in TTS. It is thus possible to directly generate audio by predicting samples one at a time in an autoregressive fashion. WaveNet leverages stacks of one-dimensional dilated convolutions with a dilation factor increasing exponentially with the layer depth, allowing for the very large receptive field and the strong nonlinearity needed to model raw audio.

Tacotron(2017) is a sequence-to-sequence model that produces a spectrogram from a sequence of characters alone, further reducing the need for domain expertise.

1. **Analysis and Implementation**

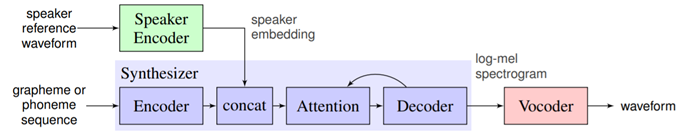
**5.1. Analysis**

The 3 stages of the framework ar as follows:

1. A speaker encoder that derives AN embedding from the short auditory communication of one speaker. The embedding may be a purposeful illustration of the voice of the speaker, such similar voices ar march on latent area.

2. A synthesizer that, conditioned on the embedding of a speaker, generates a pic from text. This model is that the fashionable Tacotron a pair of (Shen et al., 2017) period while not WaveNet.

3. A vocoder that infers AN audio wave from the spectrograms generated by the synthesizer.

****

*Fig: The SV2TTS framework during inference. The blue blocks represent a high-level view of the Tacotron architecture modified to allow conditioning on a voice.*

For the datasets of the synthesizer and vocoder, transcripts are needed and therefore the quality of the generated audio will solely be pretty much as good as that of the info. an oversized assortment of the many completely different speakers would be preferred to coach the encoder, with none sturdy demand on the background level of the audios.

**5.1.1. Encoder**

My encoder is enforced employing a Python library known as PyTorch. The model may be a 3-layer Long Short Term Memory (LSTM) with 768 hidden nodes followed by a projection layer of 256 units. this can be a comparatively little model however performs well. It additionally options a corrected Layer Unit(ReLU) layer before normalization that makes embeddings thin and a lot of explainable.

The encoder is trained for a speaker verification task. A templet for an individual is formed by account their speaker embedding, that is thought as enrollment. Then, this user identifies himself at runtime with a brief vocalization, and therefore the system compares the embedding to it of the listed speaker embeddings. on top of a given similarity threshold, the user is known.

At coaching time, the model computes the embeddings eij (1 ≤ i ≤ N, one one j ≤ M ) of M utterances of fastened length from N speakers. A speaker embedding ci comes for every speaker:



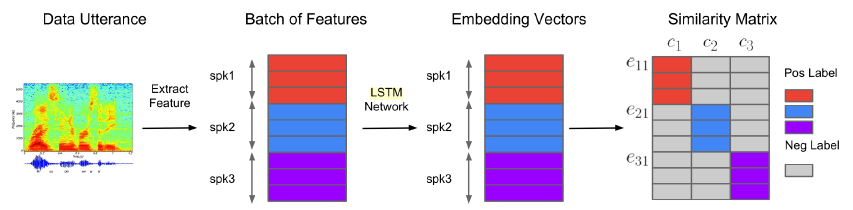
The similarity matrix Sij,k is that the results of the two-by-two comparison of all embeddings eij against each speaker embedding ck (1 ≤ k ≤ N) within the batch.

This measure is the scaled cosine similarity:

**

where w and b ar learnable parameters.

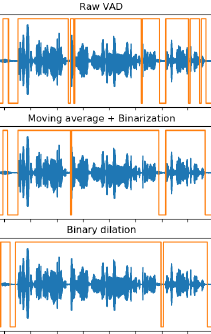
From a computing perspective, the cos similarity of 2 L2-normed vectors is just their real, thus the right hand aspect of equation a pair of. AN optimum model {is expected|is predicted|is ANticipated} to output high similarity values once an vocalization matches the speaker (i = k) and lower values elsewhere (i 6= k). To optimize during this direction, the loss is that the add of row-wise softmax losses.

**

*Fig. The construction of the similarity matrix at training time. (Wan et al., 2017).*

Note that every vocalization eij is enclosed within the centre of mass ci of an equivalent speaker once computing the loss. This creates a bias towards the proper speaker severally of the accuracy of the model, {and the|and therefore the|and additionally the} authors of SV2TTS argue that it also leaves space for trivial solutions. to forestall this, AN vocalization that's compared against its own speaker's embedding are going to be aloof from the speaker embedding.

When enrolling a speaker during a usage, many utterances ought to be loaded from every user however less than ten. As for the quantity of speakers, this range shouldn't be large since the complexness of the similarity matrix is O(N2M). If this range is just too high, the coaching time would become terribly massive.

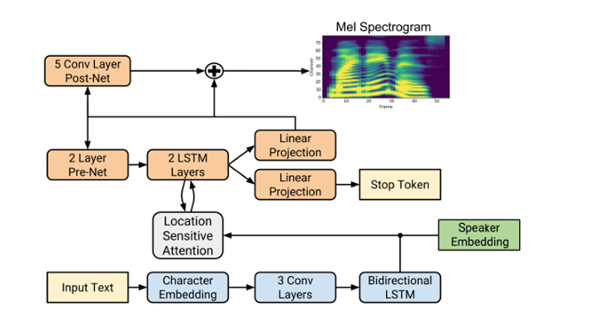


*Fig. The steps to silence removal with VAD, from top to bottom. The orange line is the binary voice flag where the upper value means that the segment is voiced, and unvoiced when lower.*

Voice Activity Detection(VAD) is completed by victimization webrtcvad python module to avoid segments of audio that ar silent once sampling partial and complete utterances. The audio is cut of the silent components that exceed a most threshold of zero.2 seconds.

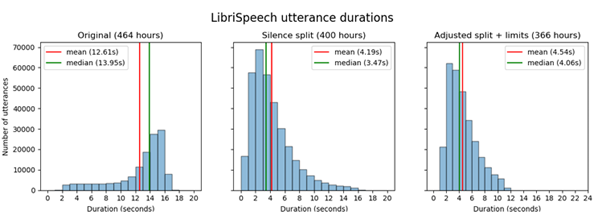
To train this model, the LibriSpeech dataset was used that contains 2 sections- “clean” and “other” wherever the clean sample section contains comparatively reduced noise. in keeping with (Wan et al., 2017), the quantity of speakers is powerfully related with the great performance of the encoder moreover because the entire framework. So, the speaker encoder ought to be trained for as several steps as possible.

**5.1.2 Synthesizer**

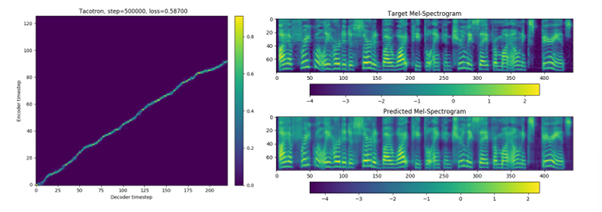
Acotron two while not Wavenet (2016) an open-source TensorFlow implementation of Tacotron 2 from that Wavenet and implement the modifications further by SV2TTS was stripped. Tacotron could be a continual sequence-to-sequence methodology that comes with text to predict a mel photograph. it's an encoder-decoder structure (not to be mistaken with the SV2TTS speaker encoder) with a location-sensitive attention mechanism. The text sequence's characters area unit foremost incorporated as vectors. Convolutional layers are added subsequently to extend the vary of one encoder frame. to provide the encoder output frames, these frames area unit versed a bifacial LSTM. this is often wherever SV2TTS changes to the architecture: each frame emitted by the Tacotron encoder is concatenated with a speaker embedding. every decoder input frame is concatenated with the previous decoder frame output versed a pre-net. concatenated vector goes through 2 unifacial LSTM layers before being projected to one mel spectrogram frame. the whole sequence of frames is versed a residual post-net before it becomes the mel spectrogram.

*Fig: The modified Tacotron architecture. The blue blocks correspond to the encoder and the orange ones to the decoder.*

The synthesizer's target mel spectrograms have additional characteristics than the speaker encoder's target mel spectrograms. {they're|they area unit} created from eighty channels and are supported a 50ms window with a twelve.5ms step. In my approach, the input texts aren't pronounced, and also the characters area unit equipped as is. However, there area unit some improvement procedures: substitution abbreviations and numerals with their full matter type, forcing all characters to code, normalizing whitespaces, and dynamical all characters to minuscule. Punctuation is also utilised, however it is not obtainable in my information.

LibriSpeech was used as a result of it offered the most effective voice cloning similarity on unseen speakers. With the audio aligned to the text, utterances on silences longer than zero.4 seconds were split. This helped the synthesizer to converge, each attributable to the removal of silences in the target spectrogram, however also because of the reduction of the median period of the utterances within the dataset, as shorter sequences provide less area for timing errors. we ensured that utterances aren't shorter than one.6 seconds, the period of partial utterances used for coaching the encoder, and not longer than eleven.25 seconds.**

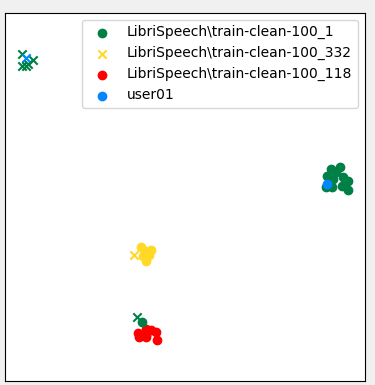
*Fig: Distribution of length of utterances in the dataset*

The model is trained in Ground Truth Aligned (GTA) mode (also referred to as a teacher-forcing mode), wherever the pre-net is fed the preceding frame of the bottom truth spectrogram instead of the expected one. it's difficult to supply any quantitative assessment of the performance of the model. **

*Fig: Tacotron producing correct outputs through informal listening tests*

It may be assessed many characteristics of the trained synthesizer exploitation Griffin-Lim (Griffin and Jae Lim, 1984) as vocoder before coaching the vocoder. Griffin-Lim is an reiterative approach that guesses the source audio signal of a spectrogram, not a machine learning model. The audio created during this manner usually retains very little of the speaker's vocal qualities, however the speech is apprehensible. Even within the presence of sophisticated or fictitious terms, the synthesizer's speech accurately fits the text. However, the prosody is often strange, with pauses in surprising places within the phrase or no pauses once they area unit anticipated. the boundaries we obligatory on the period of utterances within the dataset (1.6s -11.25s) area unit possible additionally problematic.

By computing the embeddings of synthetic speech and projecting them using UMAP alongside ground truth embeddings, it's clear that some voice qualities are lost with Griffin-Lim. The figure below shows an illustration. The synthesized embeddings clusters are similar to their respective ground truth embeddings clusters, as seen. The loss of emerging traits is additionally obvious; as an example, the synthesized utterances for the pink, red, and 2 blue speakers show lower inter-cluster variance than their ground truth counterparts. The gray and purple speakers are affected by this downside. Tacotron is usually faster than real-time.

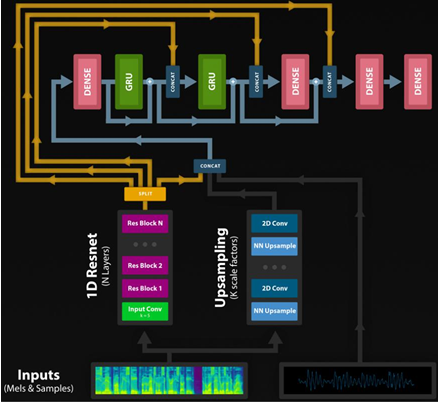
**

*Fig: UMAP projections of various speakers*

**5.1.3 Vocoder**

WaveNet has been at the heart of deep learning with audio since its release and remains state of the art when it comes to voice naturalness. It is known for being the slowest practical deep learning architecture at inference time. Google's WaveNet implementation with various improvements already generates 8000 samples per second (Kalchbrenner et al., 2018, page 2). This is in contrast with "vanilla" WaveNet which generates at 172 steps per second at best.

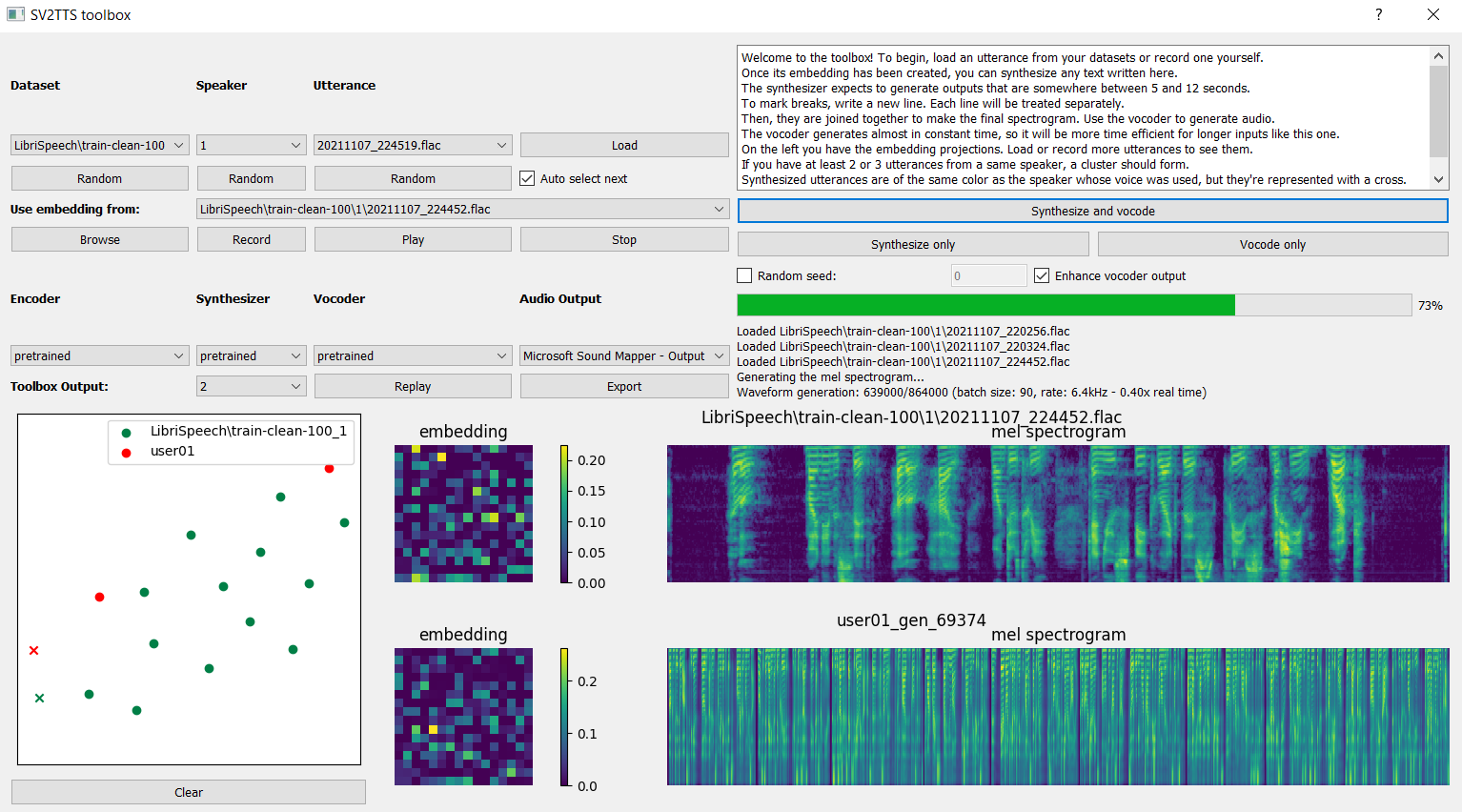
A mel spectrogram and its corresponding waveform are cut in the same number of segments. A Resnet-like model uses the spectrogram as input to generate features that will condition the layers throughout the transformation of the mel spectrogram to a waveform. Finally, two dense layers produce a distribution over discrete values that correspond to a 9-bit encoding of mu-law compounded audio.



*Fig: The alternative WaveRNN architecture.*

When dealing with short utterances, the vocoder usually runs below real-time. The inference speed is highly dependent on the number of folds in batched sampling. On my setup, this threshold is of 12.5 seconds; for utterances that are shorter than this threshold, the model will run slower than real-time on PyTorch. The authors of (Kalchbrenner et al. 2018) claim that a large sparse WaveRNN will perform better and faster than a smaller dense one. Thus, it is indicated that a pruned vocoder would produce better results in terms of speed.

**5.2 Implementation**

****

*Fig. Framework Application*

This framework is implemented in an application with a simple user interface that helps users get a quick overview of every process going on currently in the system.

The user is allowed to record an utterance or select any utterance from a dataset in their disk. After this is done, the program computes its embedding and draws its corresponding spectrogram and heatmap. However, since embeddings are unidimensional vectors, the square shape has no structural meaning about the embedding values.

When the embedding has been computed, the selected voice is used to convert the text given by the user to speech. A complete spectrogram of the output is displayed after synthesis is completed. The embedding of synthesized utterance is also generated and is projected in the UMAP as well.

1. **Result and Discussion**

In this implementation, the voice recognition section works almost flawlessly whereas the speech synthesis works adequately well. Speech similarity is quite high, whereas speech naturalness can be improved. Overall, the results were very satisfactory and above expectations.

Since this framework was run on a computer with a relatively weak GPU along with time and computational constraints, it is possible that the results did not achieve their full potential. The model was trained for a million steps, whereas the original author trained it for 50 million steps and achieved slightly better results. With an even more powerful computer, the results would likely have been even closer to human levels by achieving even greater naturalness and similarity.

1. **Conclusion and Future Research Directions**

A framework for real-time voice cloning was studied and implemented. The results exceeded expectations but could be improved with more reference speech time.

Proper datasets for audio-based projects such as this are relatively scarce. The SV2TTS team has been working on a vocoder that could clone most voices, but not uncommon ones. In the future, new frameworks could be developed that account for this overlooked aspect as well.

The importance of research in text-to-speech increases day by day since many leading technology companies aim to adopt their own artificially intelligent chatbot or personal assistants. Developments in this field are being made all the time and we would like to study and implement them in the future.

1. **References**

Corentin Jemine. Real-time Voice Cloning, 2018 <https://matheo.uliege.be/bitstream/2268.2/6801/5/s123578Jemine2019.pdf>

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation

by jointly learning to align and translate. CoRR, abs/1409.0473, 2014. [http://arxiv.org/abs/1409.0473.](https://arxiv.org/pdf/1409.0473)

Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, KyungHyun Cho, and Yoshua

Bengio. Attention-based models for speech recognition. CoRR, abs/1506.07503,

2015. <http://arxiv.org/abs/1506.07503.>

Sercan Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei

Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-

speech, 2017.

Sercan O. Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice

cloning with a few samples, 2018.

J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In

INTERSPEECH, 2018.

D. Grin and Jae Lim. Signal estimation from modified short-time fourier transform.

IEEE Transactions on Acoustics, Speech, and Signal Processing, 32(2):236-243, April 1984. ISSN 0096-3518. doi: 10.1109/TASSP.1984.1164317.