Deep Neural Networks for cloning human voice — Real world architecture

deepakvraghavan Follow

Apr 18, 2018 · 7 min read

Voice recognition is an important feature that we use extensively on a daily basis. Speech To Text (STT) conversion is used widely, some of the examples include voice commands on the phone for browsing the web or texting, voice navigation in the car, voice commands through the television remote etc. On a related note, Text to speech (TTS) systems have been in existence for a while and some of the examples include Audio books, PDF text to voice reader, etc. In all these applications, the generated voice is predetermined. The TTS pipeline converts the text to audio using the traditional Automatic Speech Recognition (ASR) modules.

With the advancements in deep neural networks, we can now take this technology to the next level. We can modify the TTS workflow to convert the text into **ANY** voice of our choice. The potential applications of this technology are mighty powerful :

- We can choose to hear an audio books in OUR own voice or the voice of a famous politician or a celebrity or mentor of our choice.
- Kids can listen to audio books that are generated using voices of their parents.
- Any voice assisted technology (GPS, smart devices in the house, generated voice on the cell phones etc) can be configured to use a voice of our preference.

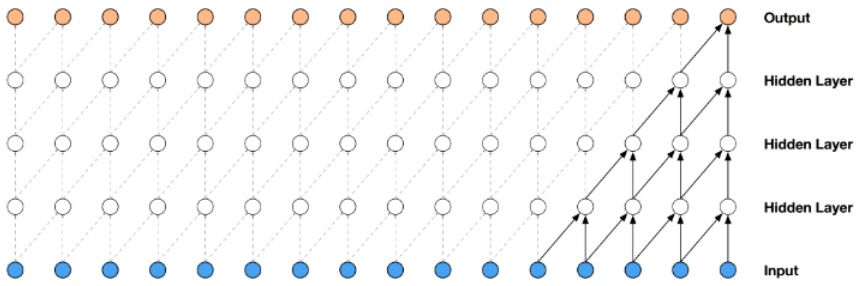
This blog post provides a brief overview of modern TTS architectures WaveNet (from Google) and Deep Voice (from Baidu) that heavily leverage Deep Neural Network architectures . At the end of the post, the author shows a voice clip sample that was generated using the Deep Voice architecture by using a corpus of his voice samples as a training set.

The year 2016 was big for the voice generation research area. Deepmind released their seminal paper titled “Wavenet” to demonstrate how voice can be generated using text effectively. This was a ground breaking invention compared to the technologies that existed until then. Traditionally, there were two approaches for TTS:

- Concatenative TTS—In this approach, the model is trained using hours of short speech fragments from a single speaker. A new speech sample is synthesized by “concatenating” these different small fragments. The drawback here is that this model is strongly tied to one voice and would take a complete retraining of the model to switch to a different voice of a speaker. Another pitfall is that unless we have a rich corpus as a training set, all the variants of emphasis or emotion or intonation (variation of pitch level) in the voice are hard to capture and generate in the voice to sound as close as possible to the human voice
- Parametric TTS—In this approach, a speech synthesizer is used that leverages signal processing algorithms and vocoders (processors used to analyze audio in frequency and time domains and generating voices). The different parameters for the synthesis are changed for each type of voice output. The drawback here is that the output has clear characteristics of “machine generated” signals which is far from a human sounding audio sample.

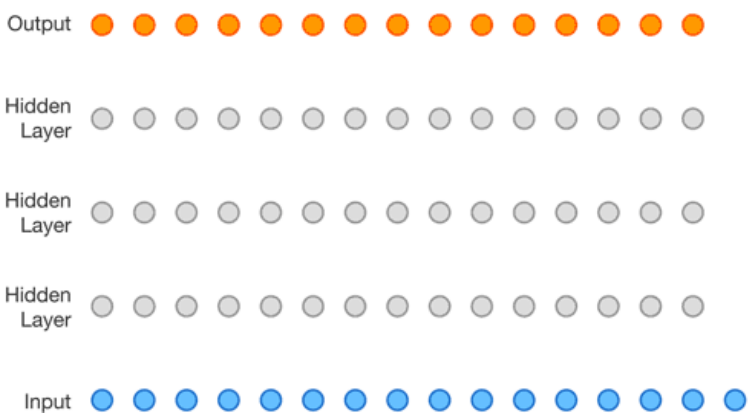
WaveNet addressed these challenges by modeling the raw audio waveform, one sample at a time. This gave the model to generate more human sounding voices—be it voice or music. WaveNet introduced the novel use of Dilated Convolutional Neural networks for voice generation.

A conventional neural network looks like the one shown below.



Normal CNN

For example, in this model an output neuron is a function of inputs which are 4-time stamps behind. This poses a challenge when trying to model human speech. Human speech needs to be analyzed for longer time intervals because in a given sentence, a word which comes after 3 or 4 words can depend on the way the current word is said because of Grammar, usage etc. For instance, the same word followed by punctuation such as question mark, exclamation mark etc can sound completely different across different sentences. In order to address this limitation, WaveNet introduced the notion of dilation in a CNN.



Dilated CNN used in WaveNet

In the above dilated CNN version, as the depth of the network increases, the ability to “remember” across extended time intervals exponentially increases. Due to the nature of the audio input of a human voice, this network closely represents human voice and is able to replicate it closely.

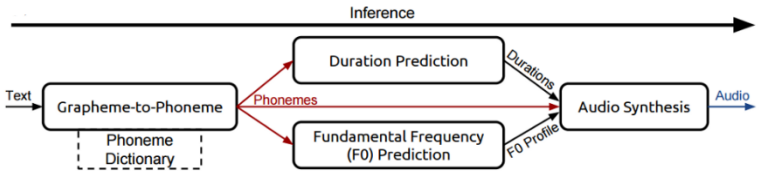
But, since WaveNet was modeling the raw audio waveform, it was relatively slow. In February 2017, Baidu released “Deep Voice” which they claimed was a production ready version for TTS generation. Deep Voice claimed a 400x speed over WaveNet, using their architecture. The main changes (improvements) in Deep Voice that enables this improvement are listed below:

- Deep Voice was the first architecture to replace all major components using Neural Networks.
- Traditional ASR architectures heavily used feature engineering and signal processing routines. Deep Voice avoids a lot of these dependencies and thus reduces the training time to just a few

hours (for end to end pipeline) compared to days and weeks of tuning in the legacy systems.

Deep Voice executes the TTS pipeline by breaking down the text to a layer of abstraction using the inference pipeline, and then training the corpus set that generated using this pipeline.

Inference pipeline



Inference pipeline

Let's use an sample input sentence, say for instance "Have a good day" to understand the process. This can be broken down into 4 key steps:

- **Converting Graphemes (Text) into Phonemes:** The words in English language sound different (or similar) even though the spellings may be similar (or different). For instance, the words "to" and "go" sound completely different although the spellings are similar. On the other hand, the words "Wait" and "Gate" sound similar even though the spellings are different. There are multiple online dictionaries we can use to get the Phonemes. CMU Dictionary is one such resource. For our example words of Wait and Gate used here, the corresponding Phonemes using CMU dictionary at http://www.speech.cs.cmu.edu/cgi-bin/cmudict are:

Wait—W EY1 T

Gate—G EY1 T

As spellings have complex pronunciations, it is recommended to use phonemes as the first layer of input for our TTS pipeline. Our sample input will turn into this output:

HH AE1 V . AH0 . G UH1 D . D EY1 .

- **Duration of Phonemes:** The way we pronounce a word depends on the phonemes which make the word and the duration of each phoneme. For instance, if we take the word “unequal” and “fun”, the phoneme for letter “u” is said for fractionally longer time in the case of “unequal” compared to it in the word fun

The second part of the input is the time duration associated with each inputs. Using this information, can write the output as follows

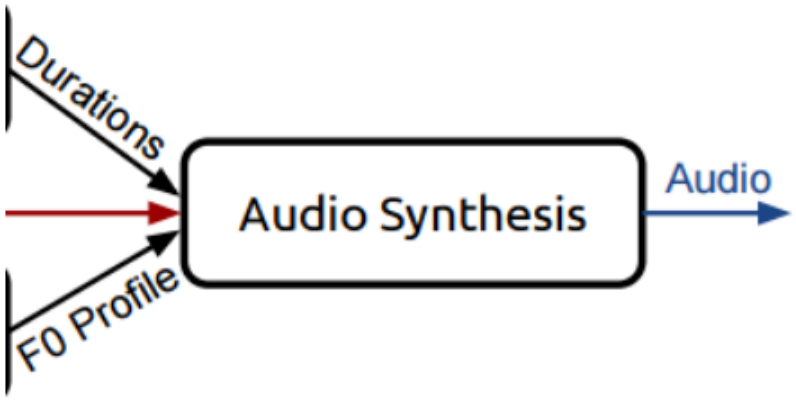
[HH (0.1s), AE1 (0.05s), V (0.05s), AH0 (0.05s), …]

- **Fundamental Frequency (F0) Prediction:** Tone and Intonation are key factors that determine how we pronounce words and phrases. Determining the fundamental frequency of a phoneme helps us to make sure that these factors are preserved in the TTS pipeline. We can use a tool or spectrometer and get the Fast Fourier Transform (FFT) to map the input signal to the Frequency domain.

Adding the F0 information, the output would look something similar to this:

[HH (180Hz), AE1 (184Hz), V (182Hz), ….]

- **Audio Synthesis:** Now that we have the the two pieces of our output (Phonemes and F0), Deep Voice employs a modified version of the above mentioned WaveNet architecture to create the voice output. The main improvement is to optimize the WaveNet implementation at higher frequency phoneme inputs. During this step, WaveNet took minutes of training to generate a second of audio output whereas Deep Voice can create the same in a fraction of a second.



Audio Synthesis to combine F0 and Phoneme durations

Result: The sound clip below is the generated output by training the Deep Neural Network using my voice and generating audio for the text quote—"Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world."

Conclusion: The audio output still does not sound 100% human, but this was created with less than an hour of training the deep neural model using my voice. But, with extensive training, we can get the outputs pretty close to human voice. The implications are exciting. This opens up also a new set of challenges in areas of security and piracy.

- If one can create a "fake" version of an audio output mimicking another human, it will be an interesting future for applications that use voice recognition for security or access.
- With advancements in this area and adequate training, we can potentially swap out one voice with another in musical compositions. This can create an interesting opportunity for software generated musical compositions which swaps a less popular vocal artist with a more popular one.