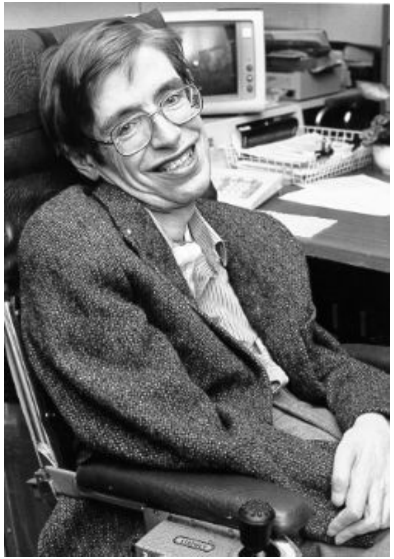
UBC MLRG 2021

WaveNet & Text to Speech

Jacques Chen

Speech Synthesis and Text to Speech



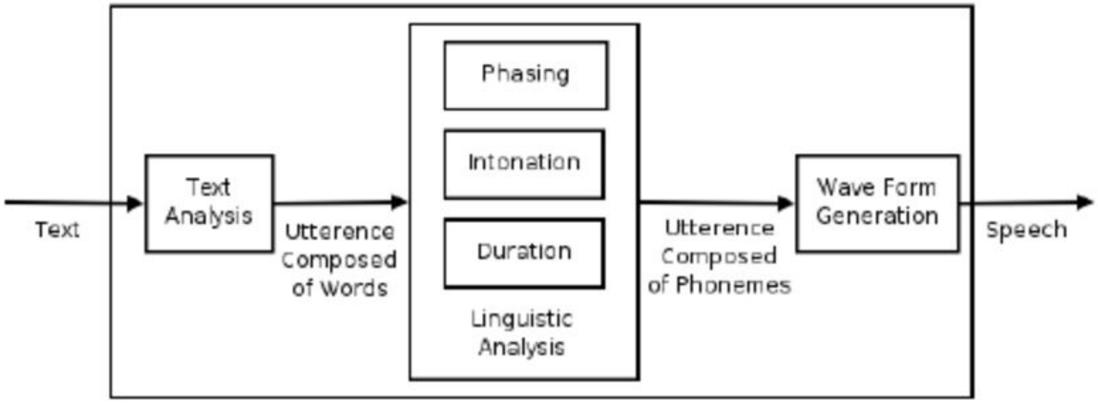
https://www.leons.ca/products/amazon-echo-dot-3rd-generation-with-alexa

https://www.theverge.com/2017/10/11/16453788/google-home-mini-smart-speaker-review

https://en.wikipedia.org/wiki/Siri#/media/File:AppleSiriIcon2017.png

https://en.wikipedia.org/wiki/Speech_synthesis#/media/File:Stephen_Hawking.StarChild.jpg

Conventional TTS



NLP Step: Take text and break down into small units of speech (Phonemes)

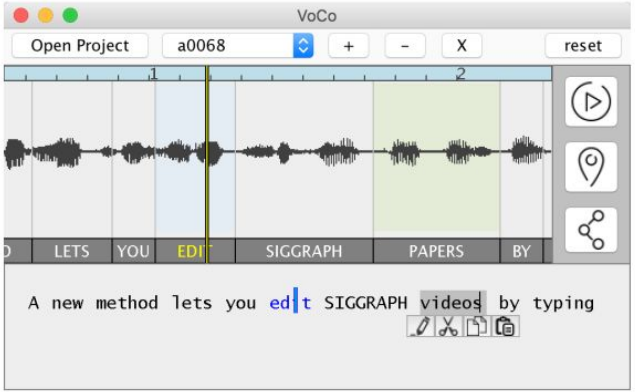
Speech Synthesis: Take phoneme sequence and generate speech waveforms

https://www.researchgate.net/figure/Block-diagram-of-a-Text-to-Speech-synthesizer_fig3_50346041

Speech Synthesis

Concatenative Models

- Take tiny samples and combine them to form speech
- Non-parametric
- Dependent on large database
- Inflexible to change
- Not natural sounding



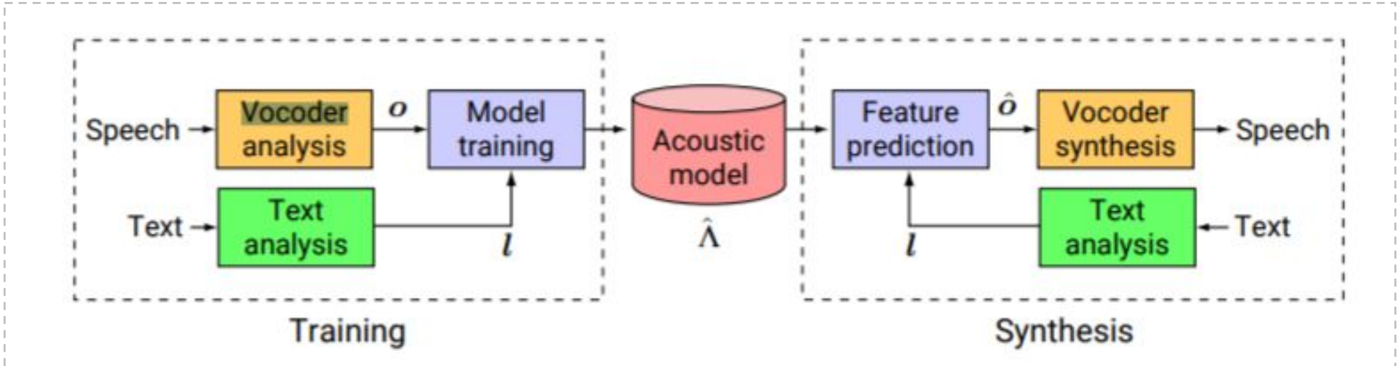
Adobe Voco, controversial "Photoshop for audio"

https://gfx.cs.princeton.edu/pubs/Jin_2017_VTI/Jin2017-VoCo-paper.pdf

Speech Synthesis

Generative Models

- Parametric
- Acoustic model could be Hidden Markov models, RNNs, Feed-forward NNs
- Still not natural sounding
- Dependent on quality of vocoders and generative models
- Receptive field is too small
- Linear filters and Gaussian assumption



$$\hat{\Lambda} = \arg\max_{\Lambda} p\left(\mathbf{o} \mid \mathbf{l}, \Lambda\right),$$

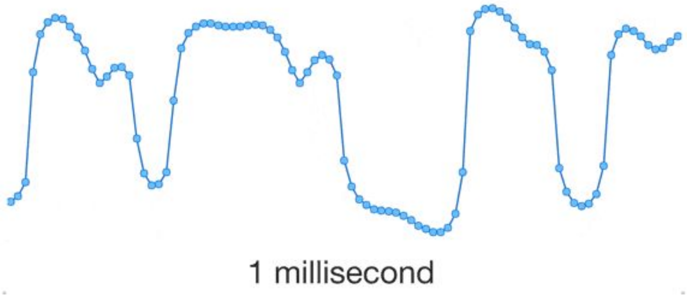
$$\hat{\mathbf{o}} = \arg\max_{\mathbf{o}} p(\mathbf{o} \mid \mathbf{l}, \hat{\Lambda}).$$

https://arxiv.org/pdf/1609.03499.pdf

Wavenet (2016) by DeepMind

- Parametric
- Autoregressive (past time-step values are inputs for current time-step)
- Handles long-range temporal dependencies
- State-of-the-art voice "naturalness"
- Useful for other applications outside of TTS
- 16 kHz sampling, input/output at each timestep is a 16-bit sequence

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

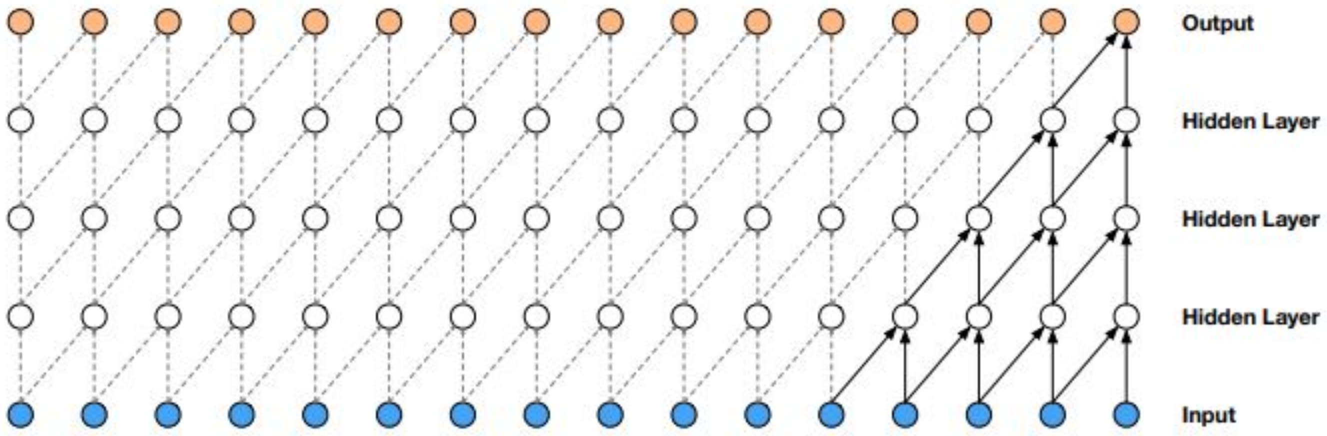


https://arxiv.org/pdf/1609.03499.pdf

https://deepmind.com/blog/article/wavenet-generative-model-raw-audio

Causal Convolutions

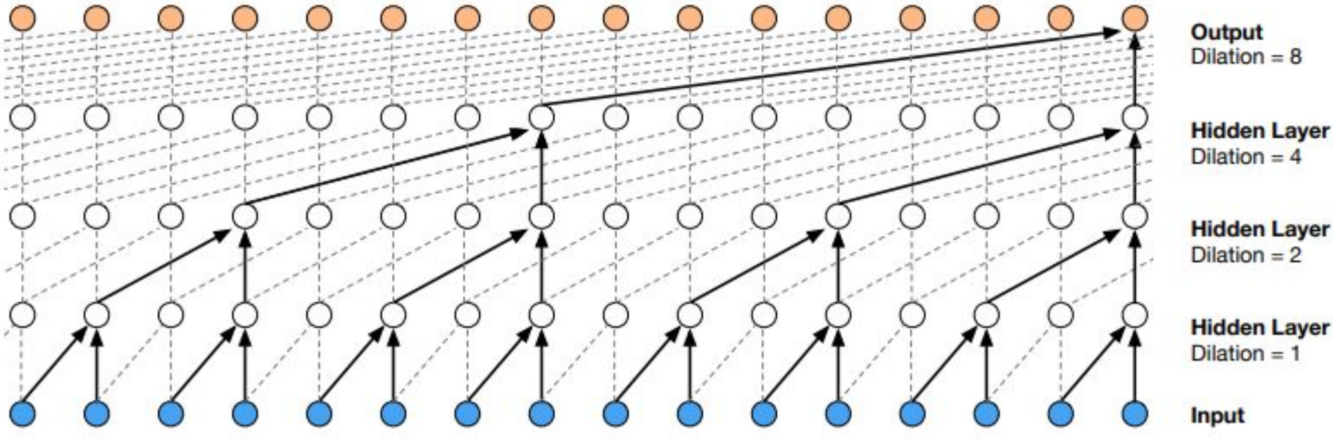
- Shift outputs by a few timesteps



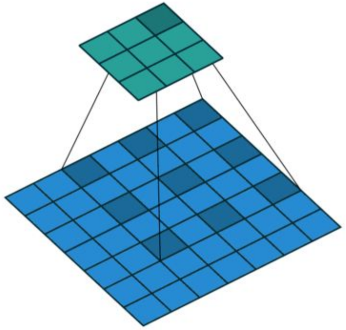
https://arxiv.org/pdf/1609.03499.pdf

Dilated Convolutions (a trous)

- Uses a dilation pattern of 1,2,4,...,512,1,2,4,...512…
- Results in exponential receptive field growth



https://arxiv.org/pdf/1609.03499.pdf



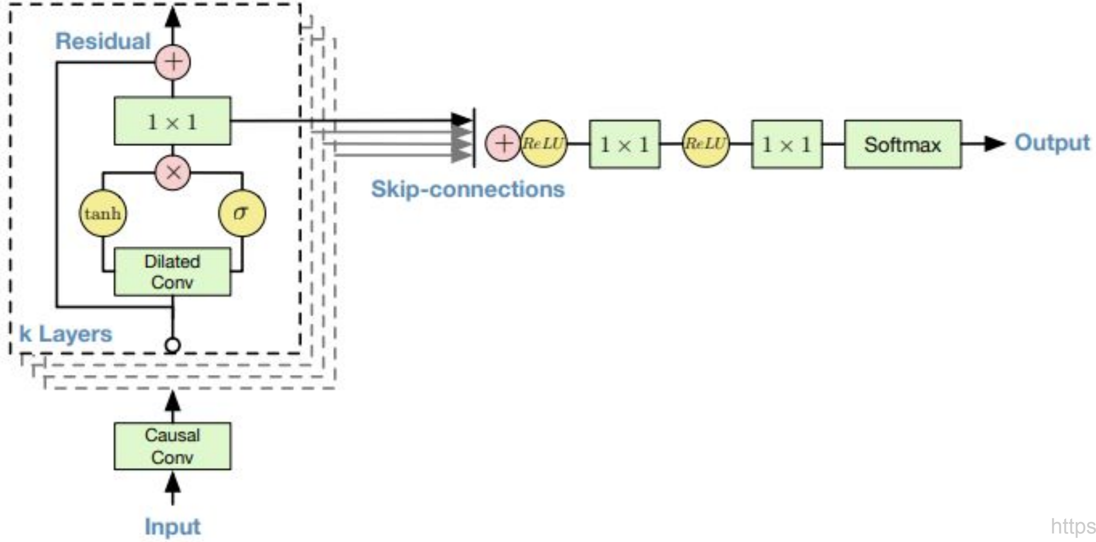
Dilated Convolution (l=2)

https://towardsdatascience.com/review-dilated-convolution-semantic-segmentation-9d5a5bd768f5

Overall Architecture

- Same gated activation unit as used in PixelCNN
- Inspired by LSTM gates

$$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x}\right) \odot \sigma\left(W_{g,k} * \mathbf{x}\right),$$



https://arxiv.org/pdf/1609.03499.pdf

Categorical Softmax

- With 16 bits per timestep, 65536 possible categories reduced to 256 with μ-law data transformation
- Common non-linear encoding used in telecommunications to reduce bit-size of audio data
- In TTS, receptive field is 240ms
- Context stacks (smaller wavenets that model longer timescales) locally condition larger Wavenet to increase its receptive field

$$f(x_t) = \text{sign}(x_t)\frac{\ln(1 + \mu|x_t|)}{\ln(1 + \mu)},$$

https://arxiv.org/pdf/1609.03499.pdf

Applied conditions

- In TTS, h would be our (local) linguistic features
- Second timeseries upsampled to map to the same resolution as the audio

$$p(\mathbf{x} \mid \mathbf{h}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1}, \mathbf{h}).$$

$$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x} + V_{f,k}^{T}\mathbf{h}\right) \odot \sigma\left(W_{g,k} * \mathbf{x} + V_{g,k}^{T}\mathbf{h}\right).$$

https://arxiv.org/pdf/1609.03499.pdf

Experiments

- Comparison test + Mean opinion score tests
- Model was also conditioned on fundamental frequency (pitch) values

https://arxiv.org/pdf/1609.03499.pdf

Speech samples	Subjective 5-scale MOS in naturalness	
	North American English	Mandarin Chinese
LSTM-RNN parametric	3.67 ± 0.098	3.79 ± 0.084
HMM-driven concatenative	3.86 ± 0.137	3.47 ± 0.108
WaveNet (L+F)	$\mathbf{4.21} \pm 0.081$	$\mathbf{4.08} \pm 0.085$
Natural (8-bit μ-law)	4.46 ± 0.067	4.25 ± 0.082
Natural (16-bit linear PCM)	4.55 ± 0.075	4.21 ± 0.071

Table 1: Subjective 5-scale mean opinion scores of speech samples from LSTM-RNN-based statistical parametric, HMM-driven unit selection concatenative, and proposed WaveNet-based speech synthesizers, 8-bit μ-law encoded natural speech, and 16-bit linear pulse-code modulation (PCM) natural speech. WaveNet improved the previous state of the art significantly, reducing the gap between natural speech and best previous model by more than 50%.

https://deepmind.com/blog/article/wavenet-generative-model-raw-audio

- Other experiments: multiple speakers, music generation, speech recognition

Drawbacks

- Fast training, super slow inference/sampling
  - Each timestep must be sequentially generated then fed as input for the next timestep
  - 0.02 seconds of audio in 1 second (using Deepmind's GPUs) for
- Not end-to-end, still dependent on NLP linguistic features (later)
- (2017) followup, **Parallel WaveNet**
- 20 seconds of audio in 1 second!
- Equivalent performance score to original WaveNet
- Now used in Google Assistant

24kHz, 16-bit lin. PCM, 65h data	
HMM-driven concatenative	4.19 ± 0.097
Autoregressive WaveNet	4.41 ± 0.069
Distilled WaveNet	4.41 ± 0.078

http://proceedings.mlr.press/v80/oord18a.html

Parallel Wavenet

- Generate all timesteps concurrently
- Inverse Autoregressive Flows (IAFs)
  - Special type of normalising flow
  - Given simple distribution $p_Z(\boldsymbol{z})$, model an invertible non-linear transformation $x_t = f(\boldsymbol{z}_{\leq t})$

$$\log p_X(\boldsymbol{x}) = \log p_Z(\boldsymbol{z}) - \log \left| \frac{d\boldsymbol{x}}{d\boldsymbol{z}} \right|,$$

  - Jacobian matrix is triangular due to time dependency, so determinant is easily calculated
  - Sampling only depends on z (fast), for Parallel Wavenet z is noise from a **logistic distribution**

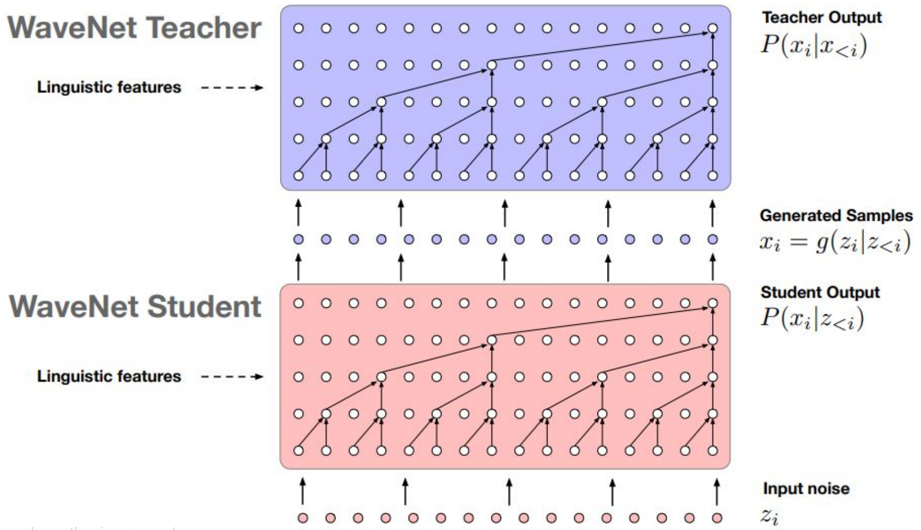
$$x_t = z_t \cdot s(\boldsymbol{z}_{<t}, \boldsymbol{\theta}) + \mu(\boldsymbol{z}_{<t}, \boldsymbol{\theta}),$$

http://proceedings.mlr.press/v80/oord18a.html

Parallel Wavenet Training

- IAFs are slow to train, fast to sample (opposite of WaveNet)
- Train our IAF model with probability density distillation
- Our "student" model learns from pretrained "Teacher" WaveNet
- In experiments, had stack of 4 IAF models

http://proceedings.mlr.press/v80/oord18a.html



https://deepmind.com/blog/article/high-fidelity-speech-synthesis-wavenet

Probability Density Distillation Loss

- Loss is the Kullback-Leibler divergence between the two models

$$D_{\mathrm{KL}}\left(P_S \| P_T\right) = \underbrace{H(P_S, P_T)}_{\text{Cross-entropy}} - \underbrace{H(P_S)}_{\text{Entropy of Student}}$$

- All these terms can be efficiently calculated after sampling from the student and calculating probabilities from the parent and student networks
- For TTS, minimize KL-divergence for same information, maximize for different (randomized) information

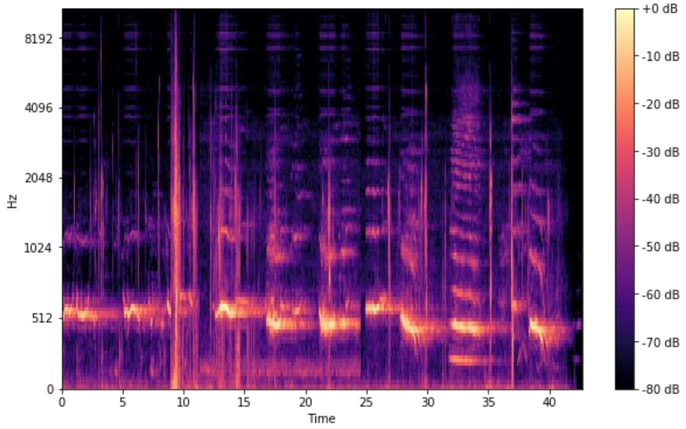
$$D_{\mathrm{KL}}\Big(P_S(\boldsymbol{c}_1)\Big\|P_T(\boldsymbol{c}_1)\Big) - \gamma D_{\mathrm{KL}}\Big(P_S(\boldsymbol{c}_1)\Big\|P_T\boldsymbol{c}_2)\Big)$$

- Additional losses to preserve proper volume and pronunciations

http://proceedings.mlr.press/v80/oord18a.html

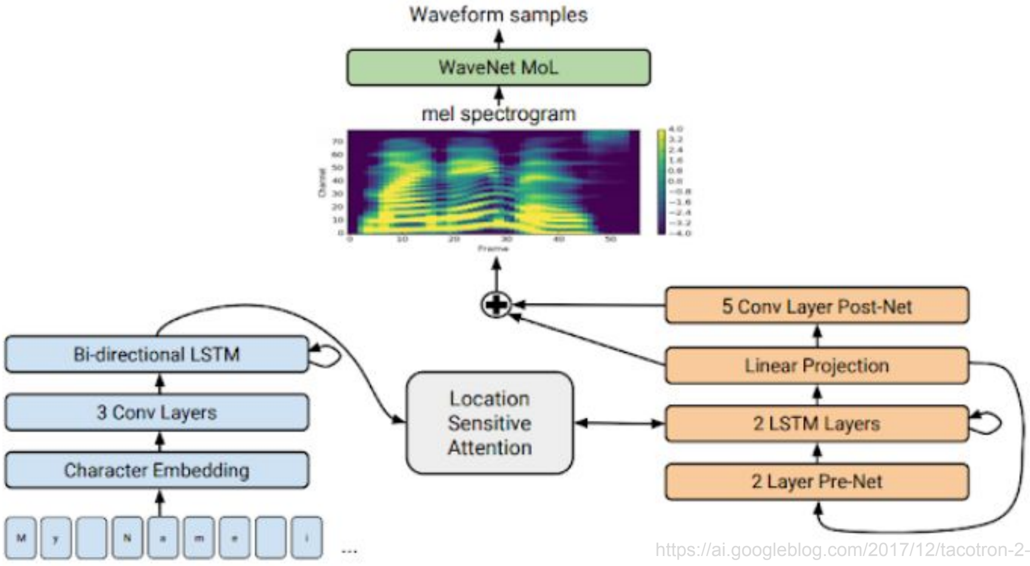
Further Improvements

- WaveNet: easy to train, hard to sample
- Parallel WaveNet: hard to train, easy to sample
- **WaveGlow** (2018): easy to train and sample
  - Uses mel-spectrogram (low level representation of audio frequencies) as input
  - Trained directly from log-likelihood of the data instead of distillation
  - Non-autoregressive
- WaveGlow and WaveNet can be conditioned on mel-spectrograms outputted by end-to-end models



https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53

End-to-end Model: Tacotron 2 (2017)



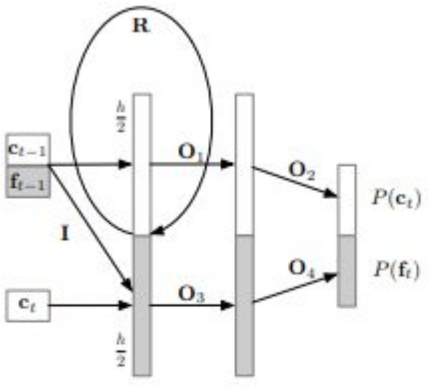
https://ai.googleblog.com/2017/12/tacotron-2-generating-human-like-speech.html

- More natural sounding output, with "volume, speed, and intonation"
- Use WaveNet as our "neural vocoder"
- Inference not fast enough for production use

Another model: Deep Voice by Baidu http://proceedings.mlr.press/v70/arik17a/arik17a.pdf

WaveRNN (2018)

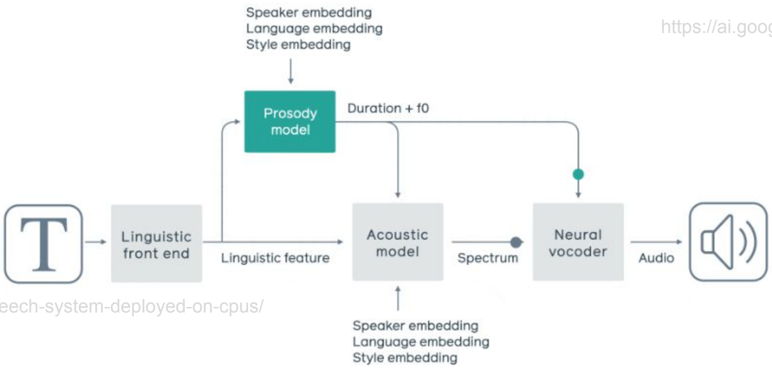
- 24 kHZ audio 4 times faster than real time on GPU (not end-to-end)
- Equal quality to original WaveNet
- Lightweight, single layer RNN
- Sparser version able to run on mobile CPU
- Used in Google Duo to preserve call quality



Modified WaveRNN used in Facebook's E2E CPU model

https://arxiv.org/abs/1802.08435

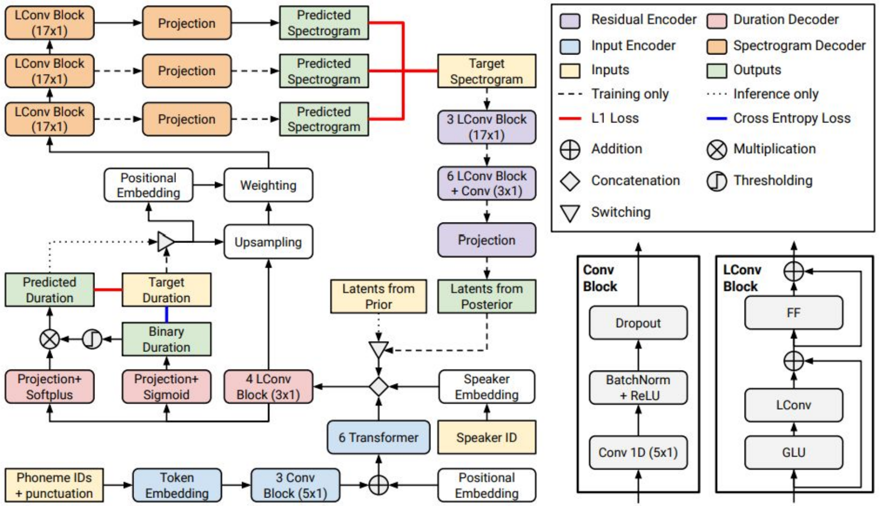
https://ai.googleblog.com/2020/04/improving-audio-quality-in-duo-with.html



https://ai.facebook.com/blog/a-highly-efficient-real-time-text-to-speech-system-deployed-on-cpus/

Parallel Tacotron (2020)

- 13 times faster inference vs Tacotron 2
- Employs transformers and lightweight convolutions for self-attention
- Non-autoregressive, uses WaveRNN to convert spectrogram to audio
- Coming soon to your local Android device?



https://arxiv.org/abs/2010.11439

Other works

- Tacotron Team (Google) + Samples

  https://google.github.io/tacotron/

- WaveNet application for speech-impaired users

  https://deepmind.com/blog/article/Using-WaveNet-technology-to-reunite-speech-impaired-users-with-their-original-voices

- Microsoft FastSpeech

  https://www.microsoft.com/en-us/research/blog/fastspeech-new-text-to-speech-model-improves-on-speed-accuracy-and-controllability/

- Siri iOS 11 on-device TTS

  https://isca-speech.org/archive/Interspeech_2017/pdfs/1798.PDF