History of Neural Vocoders for TTS

(as of 2019.10)

Seungwon Park

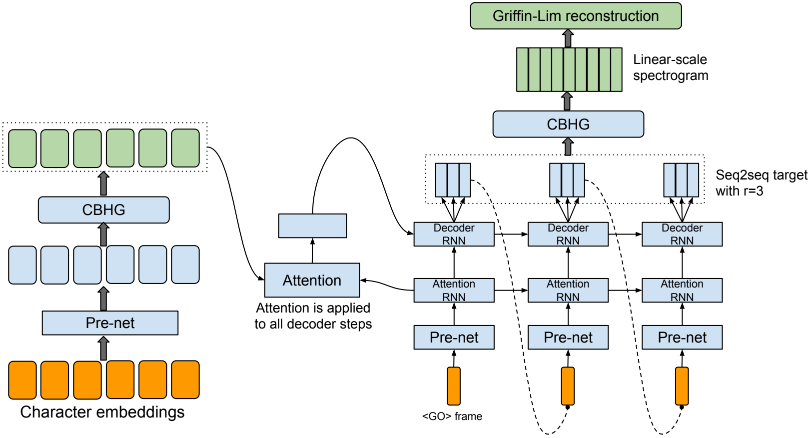
Deepest Season 6 Weekly Hosting

November 2, 2019

Introduction: Tacotron

1703.10135

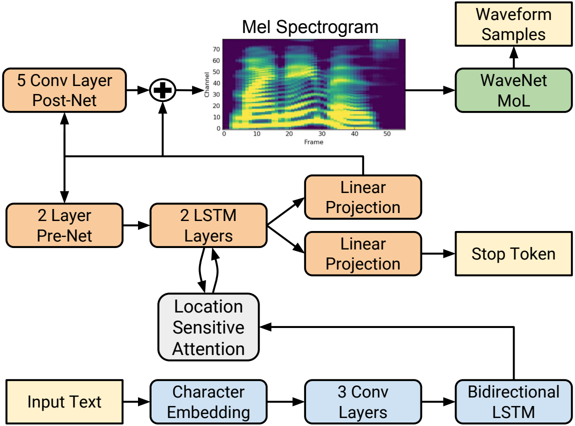
- First end-to-end model w/o F0 feature extraction



Introduction: Tacotron2

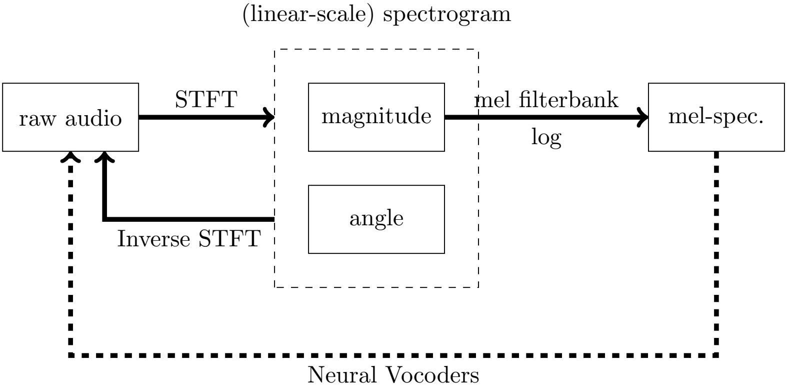
1712.05884

- Modeling mel-scale is better than linear-scale spectrogram!



Introduction: Spectrograms

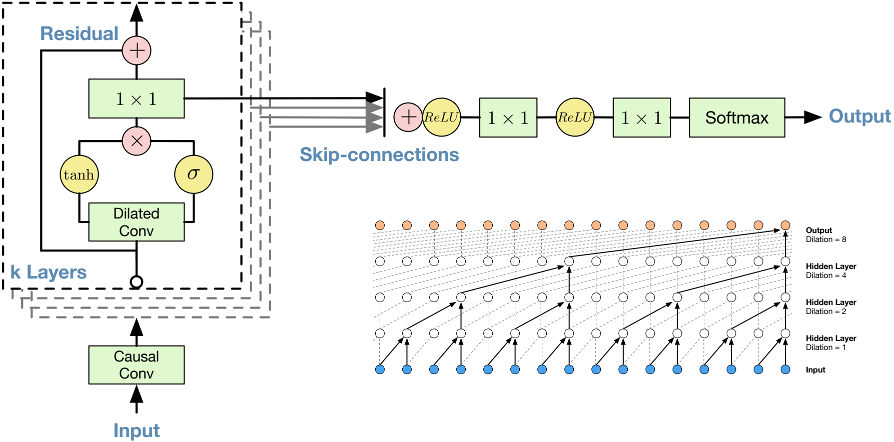
- However... mel-spectrogram is lossy compression of raw audio.
  - Hence, we need **generative models** for such inversion.



WaveNet

1609.03499

- Causal dilated conv.
- 256-way output inspired by PixelCNN



WaveNet

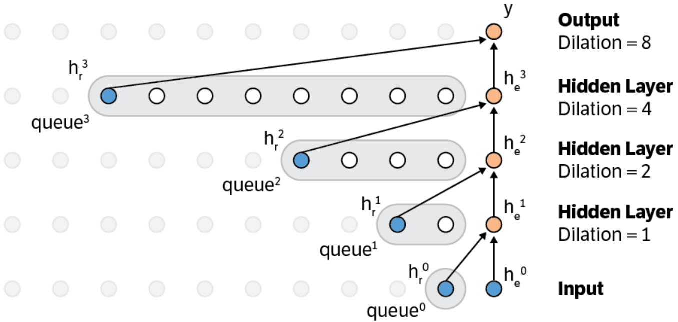
Waveform $\mathbf{x} = \{x_1, \ldots, x_T\}$ modeled with:

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

- Pros
  - Still holds SotA audio fidelity
  - Models multi-speaker speech
  - Fast training w/ teacher-forcing
- Cons
  - Horribly slow

WaveNet: powerful, but horribly slow

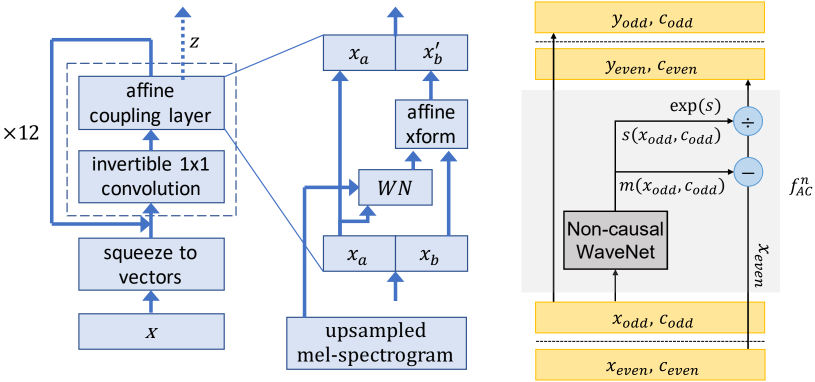
- Fast implementation w/ conv. queue (1611.09482)
- CUDA implementation: `github.com/NVIDIA/nv-wavenet`



WaveGlow / FloWaveNet

1811.00002 (ICASSP '19) / 1811.02155 (ICML '19)

- Two flow-based model with almost identical architecture



WaveGlow / FloWaveNet

Note: *WN* doesn't need to be invertible.

$$
\begin{aligned}
\boldsymbol{x}_a, \boldsymbol{x}_b &= split(\boldsymbol{x}) \\
(\log \boldsymbol{s}, \boldsymbol{t}) &= WN(\boldsymbol{x}_a, \text{mel}) \\
\boldsymbol{x}'_a &= \boldsymbol{x}_a \\
\boldsymbol{x}'_b &= \boldsymbol{s} \odot \boldsymbol{x}_b + \boldsymbol{t} \\
\boldsymbol{x}' &= concat(\boldsymbol{x}'_a, \boldsymbol{x}'_b)
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{x}'_a, \boldsymbol{x}'_b &= split(\boldsymbol{x}') \\
\boldsymbol{x}_a &= \boldsymbol{x}'_a \\
(\log \boldsymbol{s}, \boldsymbol{t}) &= WN(\boldsymbol{x}_a, \text{mel}) \\
\boldsymbol{x}_b &= \left(\boldsymbol{x}'_b - \boldsymbol{t}\right) / \boldsymbol{s} \\
\boldsymbol{x} &= concat(\boldsymbol{x}_a, \boldsymbol{x}_b)
\end{aligned}
$$

groups

x_0	x_4	x_8	x_{12}	x_{16}	x_{20}	x_{24}	x_{28}	x_{32}	x_{36}	x_{40}	$\cdots$
x_1	x_5	x_9	x_{13}	x_{17}	x_{21}	x_{25}	x_{29}	x_{33}	x_{37}	x_{41}	$\cdots$
x_2	x_6	x_{10}	x_{14}	x_{18}	x_{22}	x_{26}	x_{30}	x_{34}	x_{38}	x_{42}	$\cdots$
x_3	x_7	x_{11}	x_{15}	x_{19}	x_{23}	x_{27}	x_{31}	x_{35}	x_{39}	x_{43}	$\cdots$

WaveGlow / FloWaveNet

$$
\begin{aligned}
\log p_\theta(\boldsymbol{x}) = &- \frac{\boldsymbol{z}(\boldsymbol{x})^T \boldsymbol{z}(\boldsymbol{x})}{2\sigma^2} \\
&+ \sum_{j=0}^{\#coupling} \log \boldsymbol{s}_j(\boldsymbol{x},\ \text{mel-spectrogram}\ ) \\
&+ \sum_{k=0}^{\#conv} \log \det |\boldsymbol{W}_k|
\end{aligned}
$$

since

$$
\log p_\theta(\boldsymbol{x}) = \log p_\theta(\boldsymbol{z}) + \sum_{i=1}^{k} \log \left|\det \left(\boldsymbol{J}\left(\boldsymbol{f}_i^{-1}(\boldsymbol{x})\right)\right)\right|
$$

WaveGlow / FloWaveNet

- Pros
  - Single-stage training w/o distillation
  - Fast inference speed
- Cons
  - Requires huge amount of GPU-days to train
    - 7 days w/ 8 V100 GPUs
  - Can't model multi-speaker speech (why?)

WaveGlow / FloWaveNet

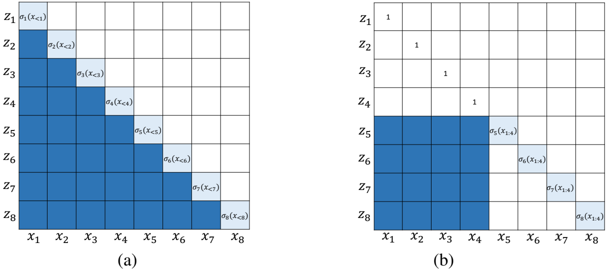


Figure 1: The Jacobian $\frac{\partial f^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}}$ of (a) an autoregressive transformation, and (b) a bipartite transformation. The blank cells are 0s and represent the independent relations between z_i and x_j. The light-blue cells are scaling variables and represent the linear dependencies between z_i and x_i. The dark-blue cells represent complex non-linear dependencies defined by neural networks.

Figure from "WaveFlow: A Compact Flow-based Model for Raw Audio"

MelGAN

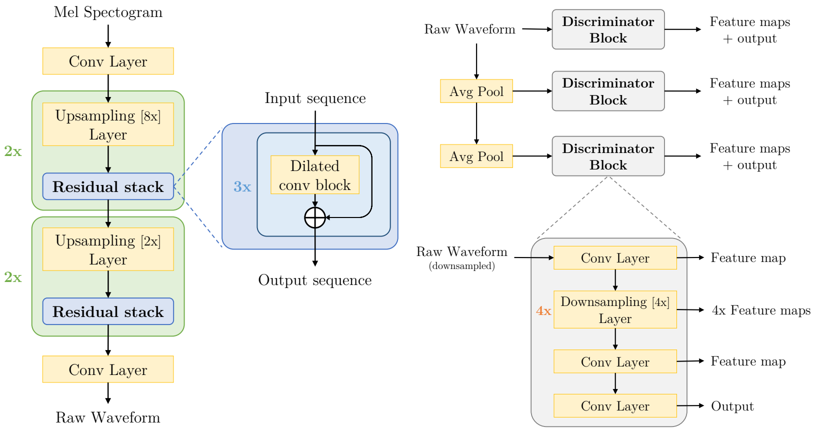
1910.06711

- Simple CNN-based GAN w/ carefully designed parameters
  - ... in non-autoregressive manner!

Table 1: Comparison of the number of parameters and the inference speed. Speed of n kHz means that the model can generate $n \times 1000$ raw audio samples per second [3].

Model	Number of parameters (in millions)	Speed on CPU (in kHz)	Speed on GPU (in kHz)
Wavenet (Shen et al., 2018)	24.7	0.0627	0.0787
Clarinet (Ping et al., 2018)	10.0	1.96	221
WaveGlow (Prenger et al., 2019)	87.9	1.58	223
MelGAN (ours)	**4.26**	**51.9**	**2500**

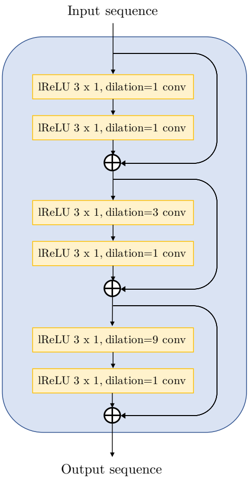
MelGAN



Mel Spectogram
Conv Layer
2x
Upsampling [8x] Layer
Residual stack
2x
Upsampling [2x] Layer
Residual stack
Conv Layer
Raw Waveform
Input sequence
3x
Dilated conv block
Output sequence
Raw Waveform
Discriminator Block
Feature maps + output
Avg Pool
Discriminator Block
Feature maps + output
Avg Pool
Discriminator Block
Feature maps + output
Raw Waveform (downsampled)
Conv Layer
Feature map
4x
Downsampling [4x] Layer
4x Feature maps
Conv Layer
Feature map
Conv Layer
Output

MelGAN

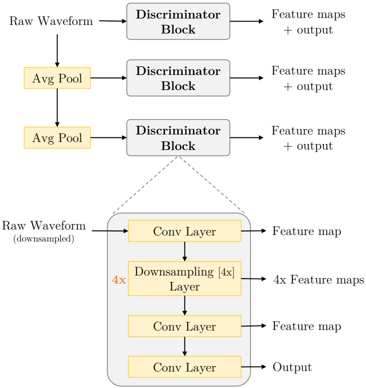
Generator



- I/O: Mel-spectrogram / Raw audio
- Upsample w/ ConvTranspose1d (as WaveGlow did)
  - $8 \times 8 \times 2 \times 2 = 256 =$ (STFT stride)
- dilation = power of kernel size: 3^i
- Do not use:
  - Latent vector z from $\mathcal{N}(0, I)$
  - Spectral norm.

MelGAN

Discriminator



- I/O: Raw audio / Feature maps
- Multi-scale modeling
- Perceptual features for G
- Least-Squares GAN objective

MelGAN

Loss function

- Discriminator:

$$\min_{D_k} \mathbb{E}_x \left[(D_k(x) - 1)^2 \right] + \mathbb{E}_{s,z} \left[D_k(G(s,z))^2 \right], \forall k = 1,2,3$$

- Generator:

$$\min_G \left(\mathbb{E}_{s,z} \left[\sum_{k=1,2,3} (D_k(G(s,z)) - 1)^2 \right] + \lambda \sum_{k=1}^{3} \mathcal{L}_{\mathrm{FM}}(G, D_k) \right)$$

where

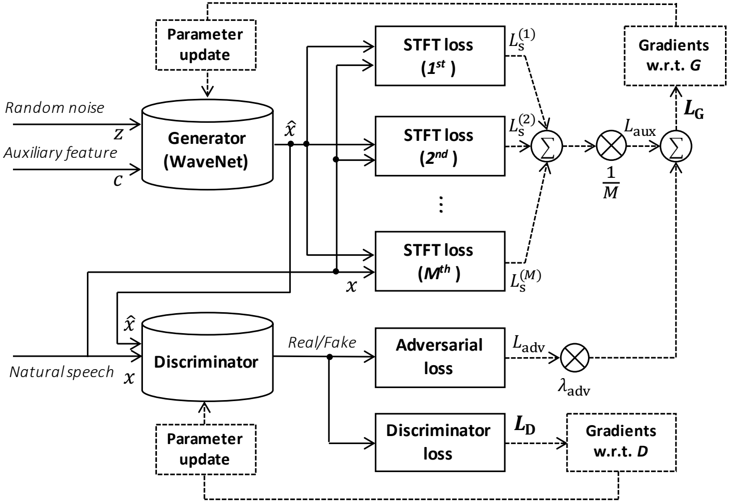
$$\mathcal{L}_{\mathrm{FM}}(G, D_k) = \mathbb{E}_{x,s \sim p_{\mathrm{data}}} \left[\sum_{i=1}^{T} \frac{1}{N_i} \left\| D_k^{(i)}(x) - D_k^{(i)}(G(s)) \right\|_1 \right]$$

MelGAN

- Pros
  - Light-weighted model w/ SotA inference speed
  - Generalizes to unseen speakers
- Cons
  - Audible artifacts of some words?
  - Painful hyper-parameter tuning
    - β values for Adam: only $(0.5, 0.9)$ works
    - Batch size affects audio fidelity: must use 16
    - Need to consider update order of G/D, batching strategy

Parallel WaveGAN

1910.11480



Overall timeline

- WaveNet: notable generative model for raw audio (2016.09)
- Tacotron2: use mel as vocoder input (2017.12)
- WaveGlow: Flow-based parallel, distillation-free model (2018.11.01)
  - FloWaveNet (2018.11.06)
- MelGAN: simple CNN-based GAN (2019.10.08)
  - Parallel WaveGAN (2019.10.25)
- (Today: 2019.11.02)

See also

- PixelCNN (1606.05328)
  - to understand background theory of WaveNet
- Parallel WaveNet (1711.10433), ClariNet (1807.07281)
  - IAF/distillation based fast models
- Behind story of FloWaveNet on Reddit (?)
- WaveFlow (OpenReview `Skeh1krtvH`)
  - Lighter version of WaveGlow, w/ good intro.
- Implementations on GitHub
  - WaveGlow: `github.com/NVIDIA/waveglow`
  - MelGAN: `github.com/descriptinc/melgan-neurips`
    - My own trial: `github.com/seungwonpark/melgan`

Audio Samples

- WaveNet w/o mel:
  `audio-samples.github.io/#section-6`
- WaveNet + Tacotron2:
  `google.github.io/tacotron/publications/tacotron2`
- WaveGlow: `nv-adlr.github.io/WaveGlow`
- MelGAN: `melgan-neurips.github.io/`

Demo

Neural TTS of MindsLab Inc. w/ Twip Inc.

- https://youtu.be/036dVJUCPRg