Generative Model-Based Text-to-Speech Synthesis

Andrew Senior (DeepMind London) Many thanks to Heiga Zen

February 23rd, 2017@Oxford

Outline

Generative TTS

Generative acoustic models for parametric TTS
- Hidden Markov models (HMMs)
- Neural networks

Beyond parametric TTS
- Learned features
- WaveNet
- End-to-end

Conclusion & future topics

Text-to-speech as sequence-to-sequence mapping

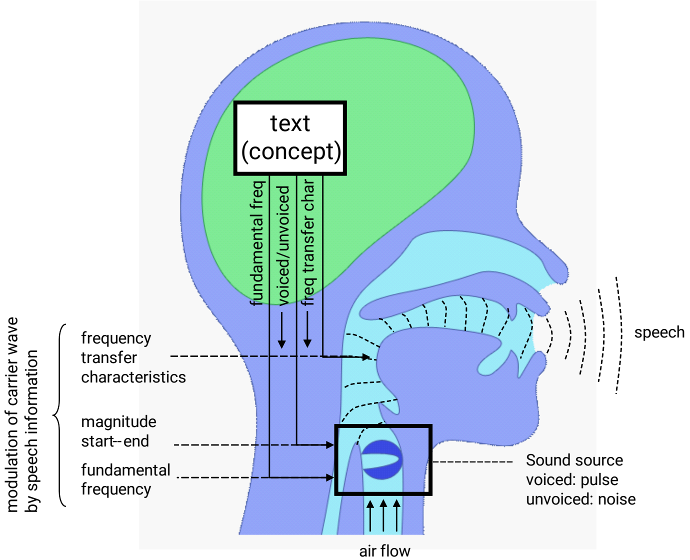
Automatic speech recognition (ASR)

 → "OK Google, directions home"

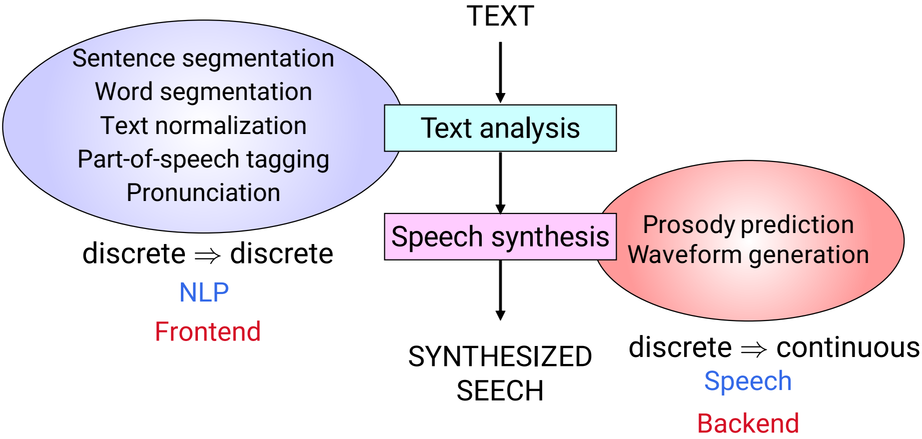
Text-to-speech synthesis (TTS)

"Take the first left" → 

Speech production process



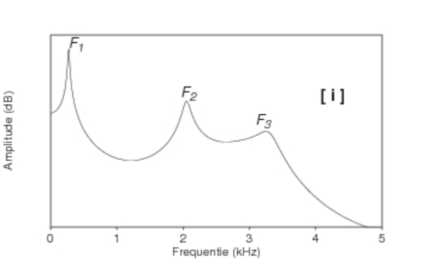
Typical flow of TTS system



Speech synthesis approaches

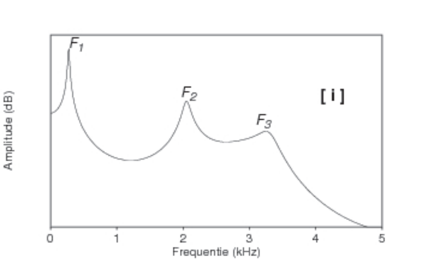
Speech synthesis approaches

Rule-based, formant synthesis [1]

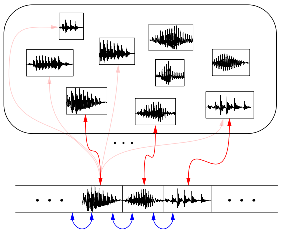


Speech synthesis approaches

Rule-based, formant synthesis [1]

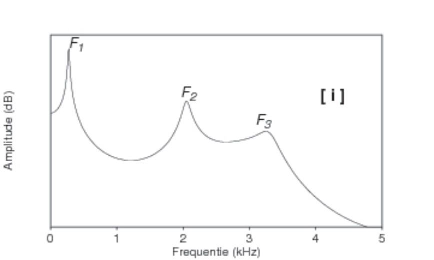


Sample-based, concatenative synthesis [2]

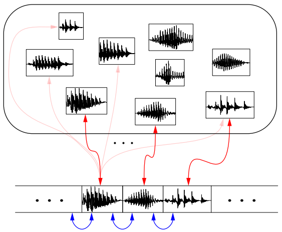


Speech synthesis approaches

Rule-based, formant synthesis [1]



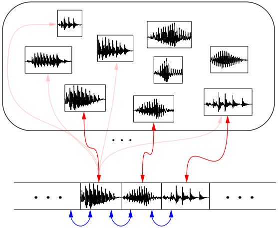
Sample-based, concatenative synthesis [2]



Model-based, generative synthesis

$p($speech=  | text="Hello, my name is Heiga Zen.")

Unit selection concatenative speech synthesis



- Build a database with wide linguistic diversity.
- Forced align and chop up into diphones.
- For a new utterance, choose units matching the diphone sequence.
- Minimize total cost by greedy search.
- Cost $= \sum_i U(i) + J(i, i-1)$
- Splice together adjacent units matching up last pitch period.

TTS databases

- Want high quality,
  - Studio recording
  - Controlled, consistent conditions
  - No background noise
  - Single (professional) speaker
- Typically read speech

TTS databases

- VCTK (Voice Cloning Tool Kit)
  - 109 native speakers of English 400 sentences. 96kHz 24 bits
  - Intended for *adaptation* of an average voice.
- Google TTS 10s of hours
- Edingburgh Merlin system
  `https://github.com/CSTR-Edinburgh/merlin`

TTS performance metrics

- TTS performance is subjective.
- Intelligibility (in noise)
- Naturalness
  - Mean Opinion Score (5 point scale)
  - A/B preference tests.
  - e.g. Amazon Mechanical Turk 100 utterances 5–7 tests per sample
  - Care needed to control for human factors.
- Objective measures
  - PESQ
  - Robust MOS

Probabilistic formulation of TTS

Random variables

$\mathcal{X}$	Speech waveforms (data)	Observed
$\mathcal{W}$	Transcriptions (data)	Observed
$\boldsymbol{w}$	Given text	Observed
$\boldsymbol{x}$	Synthesized speech	Unobserved

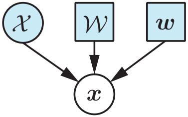
Probabilistic formulation of TTS

Random variables

$\mathcal{X}$	Speech waveforms (data)	Observed
$\mathcal{W}$	Transcriptions (data)	Observed
$\boldsymbol{w}$	Given text	Observed
$\boldsymbol{x}$	Synthesized speech	Unobserved

Synthesis

- Estimate posterior predictive distribution $\rightarrow p(\boldsymbol{x} \mid \boldsymbol{w}, \mathcal{X}, \mathcal{W})$
- Sample $\bar{\boldsymbol{x}}$ from the posterior distribution



Probabilistic formulation

Introduce auxiliary variables (*representation*) + factorize dependency

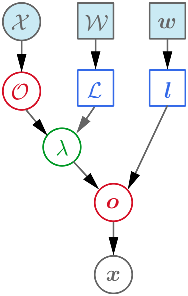
$$p(\boldsymbol{x} \mid \boldsymbol{w}, \mathcal{X}, \mathcal{W}) = \iiint \sum_{\forall \boldsymbol{l}} \sum_{\forall \mathcal{L}} \{ p(\boldsymbol{x} \mid \boldsymbol{o}) p(\boldsymbol{o} \mid \boldsymbol{l}, \lambda) p(\boldsymbol{l} \mid \boldsymbol{w})$$
$$p(\mathcal{X} \mid \mathcal{O}) p(\mathcal{O} \mid \mathcal{L}, \lambda) p(\lambda) p(\mathcal{L} \mid \mathcal{W}) / \, p(\mathcal{X}) \} d\boldsymbol{o} \, d\mathcal{O} \, d\lambda$$

where

$\mathcal{O}, \boldsymbol{o}$: *Acoustic features*

$\mathcal{L}, \boldsymbol{l}$: *Linguistic features*

λ: *Model*



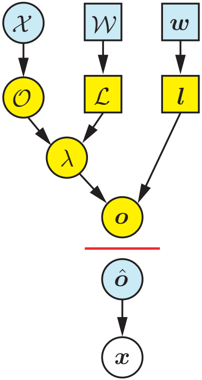
Approximation (1)

Approximate {sum & integral} by best point estimates (like MAP) [3]

$$p(\boldsymbol{x} \mid \boldsymbol{w}, \mathcal{X}, \mathcal{W}) \approx p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$$

where

$$\{\hat{\boldsymbol{o}}, \hat{\boldsymbol{l}}, \hat{\mathcal{O}}, \hat{\mathcal{L}}, \hat{\lambda}\} = \mathop{\arg\max}_{\boldsymbol{o},\boldsymbol{l},\mathcal{O},\mathcal{L},\lambda} \{ p(\boldsymbol{x} \mid \boldsymbol{o}) p(\boldsymbol{o} \mid \boldsymbol{l}, \lambda) p(\boldsymbol{l} \mid \boldsymbol{w}) p(\mathcal{X} \mid \mathcal{O}) p(\mathcal{O} \mid \mathcal{L}, \lambda) p(\lambda) p(\mathcal{L} \mid \mathcal{W}) \}$$



Approximation (2)

Joint $\rightarrow$ Step-by-step maximization [3]

$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$ Extract *acoustic features*

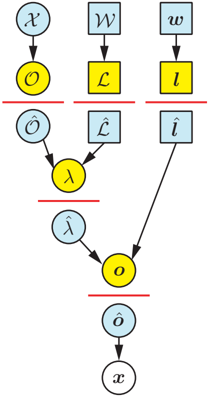
$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$ Extract *linguistic features*

$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$ Learn *mapping*

$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$ Predict *linguistic features*

$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$ Predict *acoustic features*

$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$ Synthesize waveform



Approximation (2)

Joint $\rightarrow$ Step-by-step maximization [3]

$$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$$

$$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$$

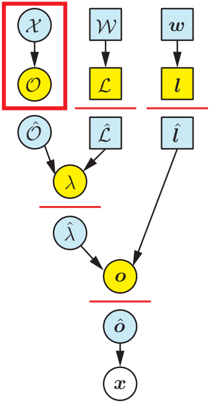
$$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$

$$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$$

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$$

$$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$$

Extract *acoustic features*



Approximation (2)

Joint $\rightarrow$ Step-by-step maximization [3]

$$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$$

Extract *acoustic features*

$$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$$

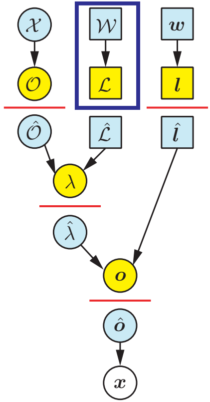
Extract *linguistic features*

$$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$

$$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$$

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$$

$$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$$



Approximation (2)

Joint $\rightarrow$ Step-by-step maximization [3]

$$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$$ Extract *acoustic features*

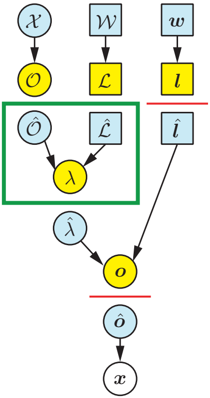
$$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$$ Extract *linguistic features*

$$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$ Learn *mapping*

$$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$$

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$$

$$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$$



Approximation (2)

Joint $\rightarrow$ Step-by-step maximization [3]

$$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$$

Extract *acoustic features*

$$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$$

Extract *linguistic features*

$$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$

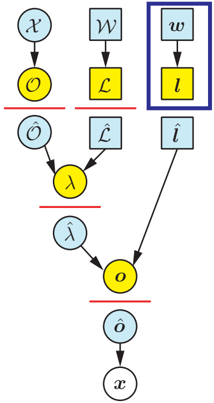
Learn *mapping*

$$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$$

Predict *linguistic features*

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$$

$$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$$



Approximation (2)

Joint $\rightarrow$ Step-by-step maximization [3]

$$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$$ Extract *acoustic features*

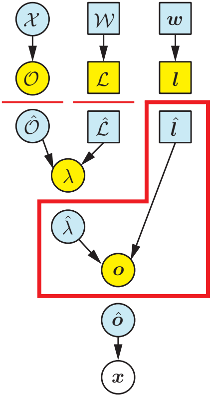
$$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$$ Extract *linguistic features*

$$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$ Learn *mapping*

$$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$$ Predict *linguistic features*

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$$ Predict *acoustic features*

$$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$$



Approximation (2)

Joint $\rightarrow$ Step-by-step maximization [3]

$$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$$ Extract *acoustic features*

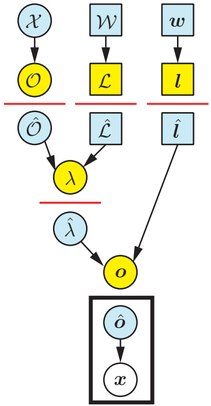
$$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$$ Extract *linguistic features*

$$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$ Learn *mapping*

$$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$$ Predict *linguistic features*

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$$ Predict *acoustic features*

$$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$$ Synthesize waveform



Approximation (2)

Joint $\rightarrow$ Step-by-step maximization [3]

$$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$$ Extract *acoustic features*

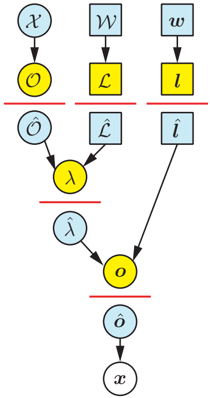
$$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$$ Extract *linguistic features*

$$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$ Learn *mapping*

$$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$$ Predict *linguistic features*

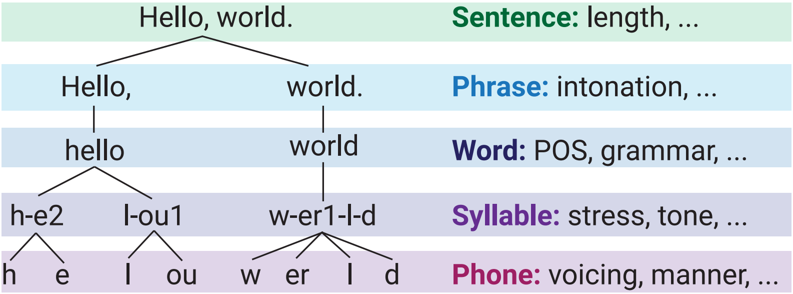
$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$$ Predict *acoustic features*

$$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$$ Synthesize waveform

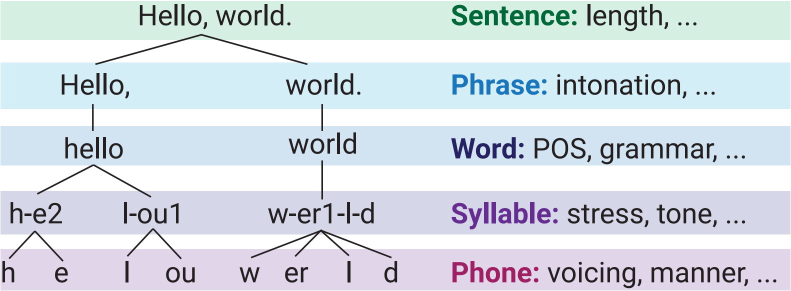


Representations: acoustic, linguistic, mapping

Representation – Linguistic features



Representation – Linguistic features



→ Based on knowledge about spoken language

- Lexicon, letter-to-sound rules
- Tokenizer, tagger, parser
- Phonology rules

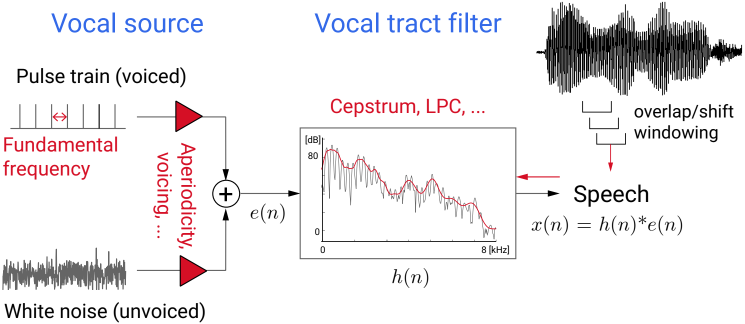
Representation – Acoustic features

Duration model

- Typically run a parametric synthesizer on frames (e.g. 5ms windows)
- Need to know how many frames each phonetic unit lasts.
- Model this separately e.g. FFNN linguistic features $\rightarrow$ duration.

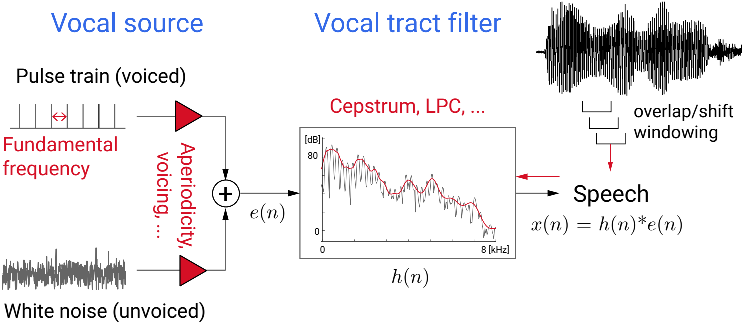
Representation – Acoustic features

Piece-wise stationary, source-filter generative model $p(x \mid o)$



Representation – Acoustic features

Piece-wise stationary, source-filter generative model $p(x \mid o)$



→ Needs to solve inverse problem

- Estimate parameters from signals
- Use estimated parameters (e.g., cepstrum) as acoustic features

Representation – Mapping

Rule-based, formant synthesis [1]

$$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$$ Vocoder analysis

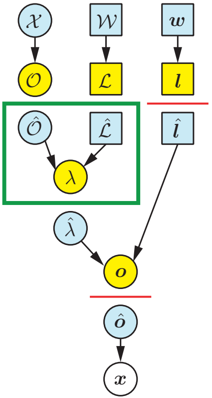
$$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$$ Text analysis

$$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$ **Extract rules**

$$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$$ Text analysis

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$$ **Apply rules**

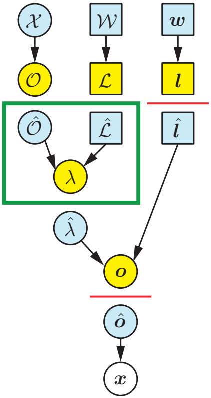
$$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$$ Vocoder synthesis



Representation – Mapping

Rule-based, formant synthesis [1]

$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$	Vocoder analysis
$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$	Text analysis
$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$	**Extract rules**
$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$	Text analysis
$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$	**Apply rules**
$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$	Vocoder synthesis



$\rightarrow$ Hand-crafted rules on knowledge-based features

Representation – Mapping

HMM-based [4], statistical parametric synthesis [5]

$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$ Vocoder analysis

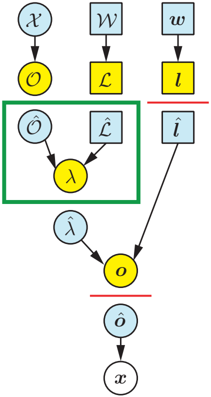
$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$ Text analysis

$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$ **Train HMMs**

$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$ Text analysis

$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$ **Parameter generation**

$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$ Vocoder synthesis



Representation – Mapping

HMM-based [4], statistical parametric synthesis [5]

$$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O})$$ Vocoder analysis

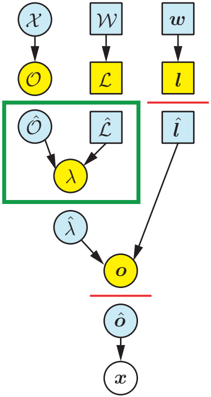
$$\hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W})$$ Text analysis

$$\hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$ **Train HMMs**

$$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l} \mid \boldsymbol{w})$$ Text analysis

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{l}}, \hat{\lambda})$$ **Parameter generation**

$$\bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\hat{\boldsymbol{o}}) = p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$$ Vocoder synthesis



→ Replace rules by HMM-based generative acoustic model

Outline

Generative TTS

Generative acoustic models for parametric TTS

- Hidden Markov models (HMMs)
- Neural networks

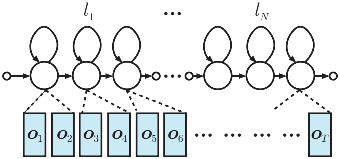
Beyond parametric TTS

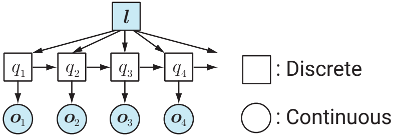
- Learned features
- WaveNet
- End-to-end

Conclusion & future topics

HMM-based generative acoustic model for TTS

- Context-dependent subword HMMs
- Decision trees to cluster & tie HMM states → *interpretable*





$$p(\boldsymbol{o} \mid \boldsymbol{l}, \lambda) = \sum_{\forall \boldsymbol{q}} \prod_{t=1}^{T} p(\boldsymbol{o}_t \mid q_t, \lambda) P(\boldsymbol{q} \mid \boldsymbol{l}, \lambda)$$ q_t: hidden state at t

$$= \sum_{\forall \boldsymbol{q}} \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t}) P(\boldsymbol{q} \mid \boldsymbol{l}, \lambda)$$

HMM-based generative acoustic model for TTS

- Non-smooth, step-wise statistics
  → Smoothing is essential
- Difficult to use high-dimensional acoustic features (e.g., raw spectra)
  → Use low-dimensional features (e.g., cepstra)
- Data fragmentation
  → Ineffective, local representation

A lot of research work have been done to address these issues

Outline

Generative TTS

Generative acoustic models for parametric TTS

- Hidden Markov models (HMMs)
- Neural networks

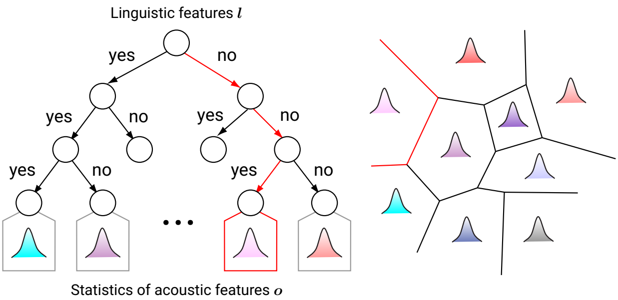
Beyond parametric TTS

- Learned features
- WaveNet
- End-to-end

Conclusion & future topics

Alternative acoustic model

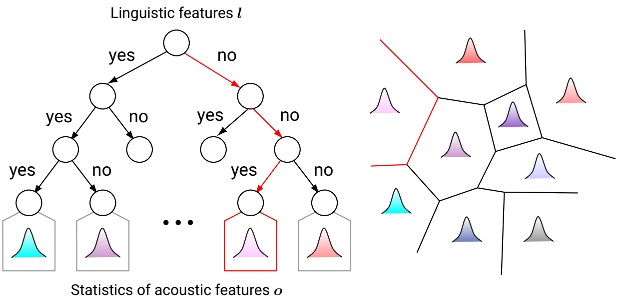
HMM: Handle variable length & alignment
Decision tree: Map linguistic $\rightarrow$ acoustic



Regression tree: linguistic features $\rightarrow$ Stats. of acoustic features

Alternative acoustic model

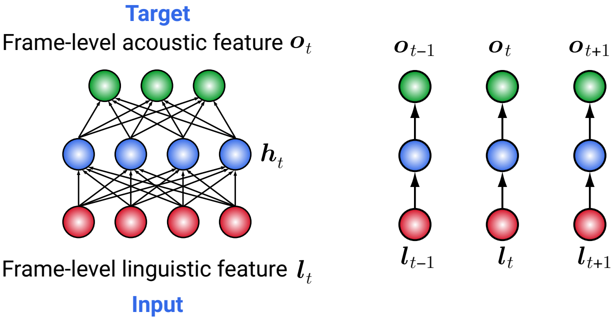
HMM: Handle variable length & alignment
Decision tree: Map linguistic → acoustic



Regression tree: linguistic features → Stats. of acoustic features

Replace the tree w/ a general-purpose regression model
→ **Artificial neural network**

FFNN-based acoustic model for TTS [6]

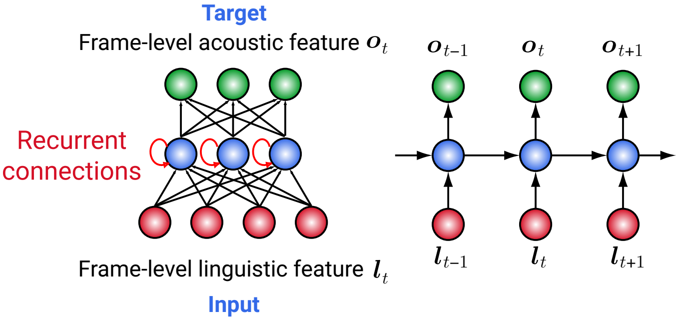


$$\boldsymbol{h}_t = g\left(\boldsymbol{W}_{hl}\boldsymbol{l}_t + \boldsymbol{b}_h\right) \qquad \hat{\lambda} = \arg\min_{\lambda} \sum_t \|\boldsymbol{o}_t - \hat{\boldsymbol{o}}_t\|_2$$

$$\hat{\boldsymbol{o}}_t = \boldsymbol{W}_{oh}\boldsymbol{h}_t + \boldsymbol{b}_o \qquad \lambda = \{\boldsymbol{W}_{hl}, \boldsymbol{W}_{oh}, \boldsymbol{b}_h, \boldsymbol{b}_o\}$$

$\hat{\boldsymbol{o}}_t \approx \mathbb{E}\left[\boldsymbol{o}_t \mid \boldsymbol{l}_t\right] \rightarrow$ Replace decision trees & Gaussian distributions

RNN-based acoustic model for TTS [7]



$$\boldsymbol{h}_t = g\left(\boldsymbol{W}_{hl}\boldsymbol{l}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h\right) \qquad \hat{\lambda} = \arg\min_{\lambda} \sum_t \|\boldsymbol{o}_t - \hat{\boldsymbol{o}}_t\|_2$$

$$\hat{\boldsymbol{o}}_t = \boldsymbol{W}_{oh}\boldsymbol{h}_t + \boldsymbol{b}_o \qquad \lambda = \{\boldsymbol{W}_{hl}, \boldsymbol{W}_{hh}, \boldsymbol{W}_{oh}, \boldsymbol{b}_h, \boldsymbol{b}_o\}$$

FFNN: $\hat{\boldsymbol{o}}_t \approx \mathbb{E}\left[\boldsymbol{o}_t \mid \boldsymbol{l}_t\right]$ RNN: $\hat{\boldsymbol{o}}_t \approx \mathbb{E}\left[\boldsymbol{o}_t \mid \boldsymbol{l}_1, \ldots, \boldsymbol{l}_t\right]$

NN-based generative acoustic model for TTS

- Non-smooth, step-wise statistics
  → RNN predicts smoothly varying acoustic features [7, 8]
- Difficult to use high-dimensional acoustic features (e.g., raw spectra)
  → Layered architecture can handle high-dimensional features [9]
- Data fragmentation
  → Distributed representation [10]

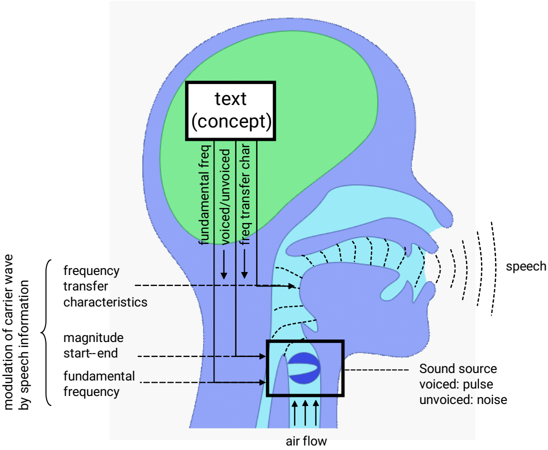
NN-based generative acoustic model for TTS

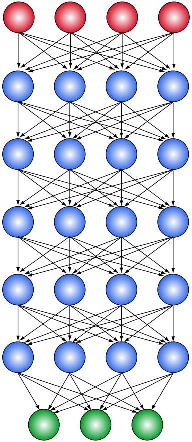
- Non-smooth, step-wise statistics
  → RNN predicts smoothly varying acoustic features [7, 8]
- Difficult to use high-dimensional acoustic features (e.g., raw spectra)
  → Layered architecture can handle high-dimensional features [9]
- Data fragmentation
  → Distributed representation [10]

NN-based approach is now mainstream in research & products

- Models: FFNN [6], MDN [11], RNN [7], Highway network [12], GAN [13]
- Products: e.g., Google [14]

NN-based generative model for TTS





Text → Linguistic → (Articulatory) → Acoustic → Waveform

Outline

Generative TTS

Generative acoustic models for parametric TTS

- Hidden Markov models (HMMs)
- Neural networks

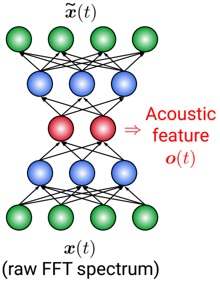
Beyond parametric TTS

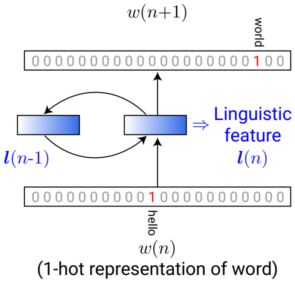
- Learned features
- WaveNet
- End-to-end

Conclusion & future topics

Knowledge-based features → Learned features

Unsupervised feature learning





- Speech: auto-encoder at FFT spectra [9, 15] → positive results
- Text: word [16], phone & syllable [17] → less positive

Relax approximation

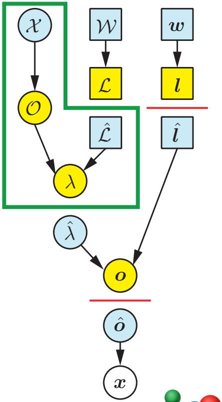
Joint acoustic feature extraction & model training

Two-step optimization $\rightarrow$ **Joint optimization**

$$\begin{cases} \hat{\mathcal{O}} = \arg\max_{\mathcal{O}} p(\mathcal{X} \mid \mathcal{O}) \\ \hat{\lambda} = \arg\max_{\lambda} p(\hat{\mathcal{O}} \mid \hat{\mathcal{L}}, \lambda) p(\lambda) \end{cases}$$

$$\Downarrow$$

$$\{\hat{\lambda}, \hat{\mathcal{O}}\} = \arg\max_{\lambda, \mathcal{O}} p(\mathcal{X} \mid \mathcal{O}) p(\mathcal{O} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$



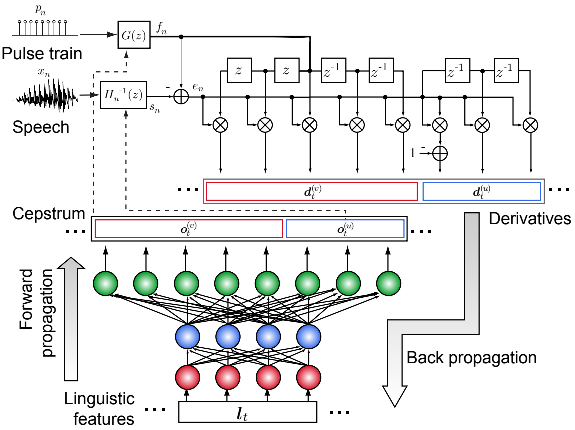
Joint source-filter & acoustic model optimization

- HMM [18, 19, 20]
- NN [21, 22]

Relax approximation

Joint acoustic feature extracion & model training

Mixed-phase cepstral analysis + LSTM-RNN [22]



Outline

Relax approximation

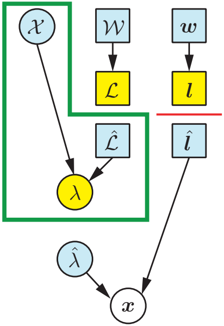
Direct mapping from linguistic to waveform

No explicit acoustic features

$$\{\hat{\lambda}, \hat{\mathcal{O}}\} = \arg\max_{\lambda,\mathcal{O}} p(\mathcal{X} \mid \mathcal{O}) p(\mathcal{O} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$

$$\Downarrow$$

$$\hat{\lambda} = \arg\max_{\lambda} p(\mathcal{X} \mid \hat{\mathcal{L}}, \lambda) p(\lambda)$$



Generative models for raw audio

- LPC [23]
- WaveNet [24]
- SampleRNN [25]

WaveNet: A generative model for raw audio

Autoregressive (AR) modelling of speech signals

$$\boldsymbol{x} = \{x_0, x_1, \ldots, x_{N-1}\} \quad : \text{raw waveform}$$

$$p(\boldsymbol{x} \mid \lambda) = p(x_0, x_1, \ldots, x_{N-1} \mid \lambda) = \prod_{n=0}^{N-1} p(x_n \mid x_0, \ldots, x_{n-1,} \lambda)$$

WaveNet: A generative model for raw audio

Autoregressive (AR) modelling of speech signals

$$\boldsymbol{x} = \{x_0, x_1, \ldots, x_{N-1}\} \quad : \text{raw waveform}$$

$$p(\boldsymbol{x} \mid \lambda) = p(x_0, x_1, \ldots, x_{N-1} \mid \lambda) = \prod_{n=0}^{N-1} p(x_n \mid x_0, \ldots, x_{n-1,} \lambda)$$

WaveNet [24]

$\rightarrow p(x_n \mid x_0, \ldots, x_{n-1}, \lambda)$ is modeled by *convolutional NN*

WaveNet: A generative model for raw audio

Autoregressive (AR) modelling of speech signals

$$\boldsymbol{x} = \{x_0, x_1, \ldots, x_{N-1}\} \quad : \text{raw waveform}$$

$$p(\boldsymbol{x} \mid \lambda) = p(x_0, x_1, \ldots, x_{N-1} \mid \lambda) = \prod_{n=0}^{N-1} p(x_n \mid x_0, \ldots, x_{n-1}, \lambda)$$

WaveNet [24]

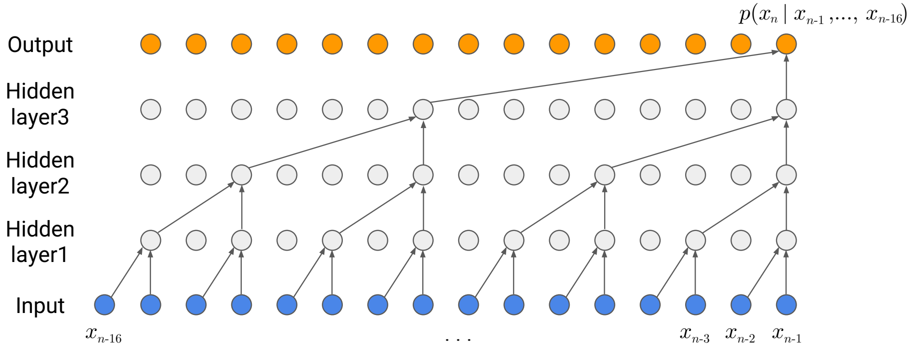
$\rightarrow p(x_n \mid x_0, \ldots, x_{n-1}, \lambda)$ is modeled by *convolutional NN*

Key components

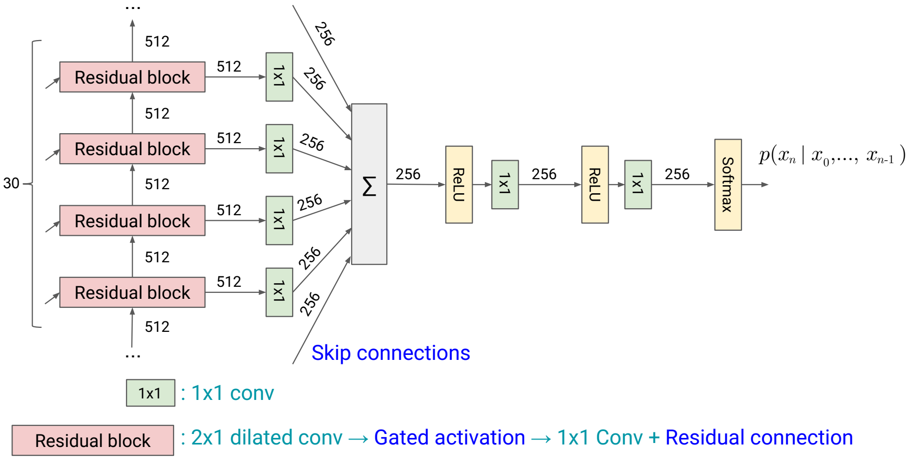
- *Causal dilated convolution*: capture long-term dependency
- *Gated convolution + residual + skip*: powerful non-linearity
- *Softmax at output*: classification rather than regression

100ms in 16kHz sampling = 1,600 time steps
→ Too long to be captured by normal RNN/LSTM

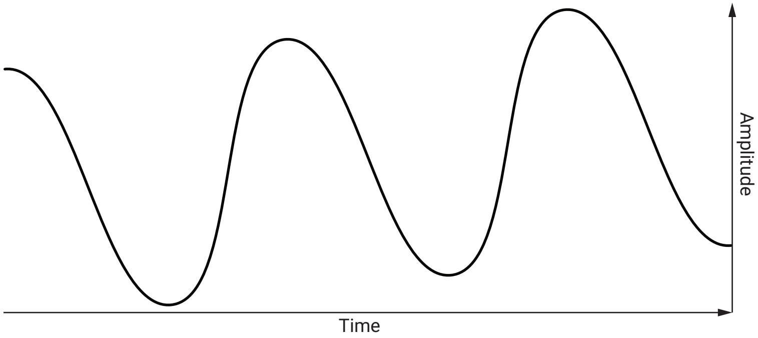
Dilated convolution
Exponentially increase receptive field size w.r.t. # of layers



WaveNet – Non-linearity

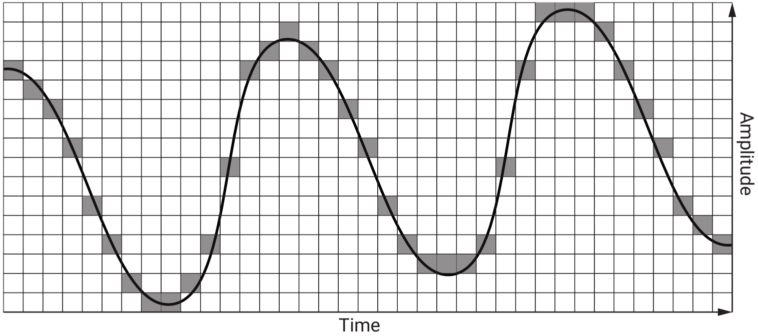


WaveNet – Softmax



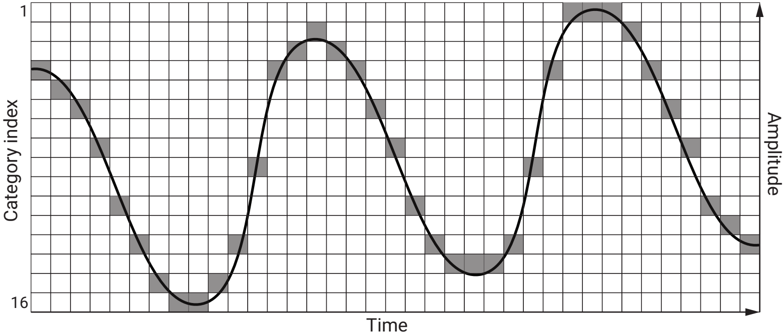
Analog audio signal

WaveNet – Softmax



Sampling & Quantization

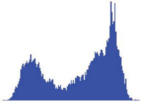
WaveNet – Softmax



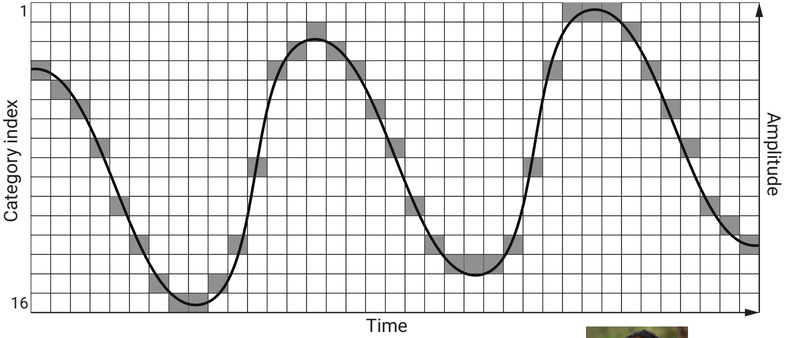
Categorical distribution → Histogram

- Unimodal
- Multimodal
- Skewed

...



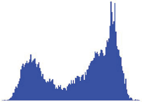
WaveNet – Softmax



Categorical distribution → Histogram

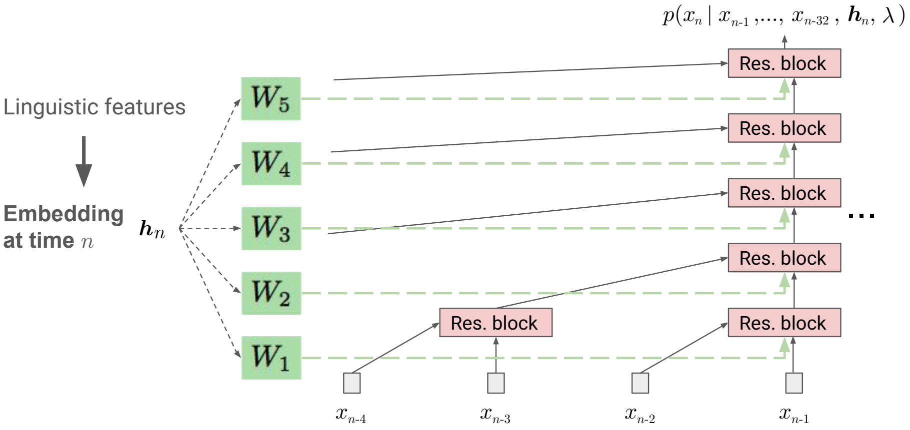
- Unimodal
- Multimodal
- Skewed

...



Prof. D. Jurafsky - *"Now TTS is the same problem as language modeling!"*

WaveNet – Conditional modelling



WaveNet vs conventional audio generative models

Assumptions in conventional audio generative models [23, 26, 27, 22]

- Stationary process w/ fixed-length analysis window
  → Estimate model within 20–30ms window w/ 5–10 shift
- Linear, time-invariant filter within a frame
  → Relationship between samples can be non-linear
- Gaussian process
  → Assumes speech signals are normally distributed

WaveNet

- Sample-by-saple, non-linear, capable to take additional inputs
- Arbitrary-shaped signal distribution

SOTA subjective naturalness w/ WaveNet-based TTS [24]

HMM LSTM Concatenative WaveNet

Outline

Generative TTS

Generative acoustic models for parametric TTS

- Hidden Markov models (HMMs)
- Neural networks

Beyond parametric TTS

- Learned features
- WaveNet
- End-to-end

Conclusion & future topics

Relax approximation

Towards Bayesian end-to-end TTS

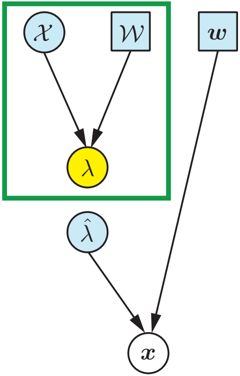
Integrated end-to-end

$$\begin{cases} \hat{\mathcal{L}} = \arg\max_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{W}) \\ \hat{\lambda} = \arg\max_{\lambda} p(\mathcal{X} \mid \hat{\mathcal{L}}, \lambda) p(\lambda) \end{cases}$$

$$\Downarrow$$

$$\hat{\lambda} = \arg\max_{\lambda} p(\mathcal{X} \mid \mathcal{W}, \lambda) p(\lambda)$$

Text analysis is integrated to model



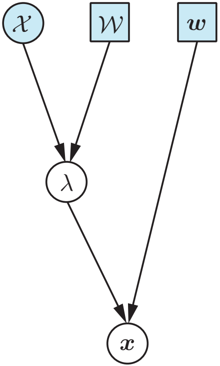
Relax approximation
Towards Bayesian end-to-end TTS

Bayesian end-to-end

$$\begin{cases} \hat{\lambda} = \arg\max_{\lambda} p(\mathcal{X} \mid \mathcal{W}, \lambda) p(\lambda) \\ \bar{\boldsymbol{x}} \sim f_{\boldsymbol{x}}(\boldsymbol{w}, \hat{\lambda}) = p(\boldsymbol{x} \mid \boldsymbol{w}, \hat{\lambda}) \end{cases}$$

$$\Downarrow$$

$$\begin{aligned} \bar{\boldsymbol{x}} &\sim f_{\boldsymbol{x}}(\boldsymbol{w}, \mathcal{X}, \mathcal{W}) = p(\boldsymbol{x} \mid \boldsymbol{w}, \mathcal{X}, \mathcal{W}) \\ &= \int p(\boldsymbol{x} \mid \boldsymbol{w}, \lambda) p(\lambda \mid \mathcal{X}, \mathcal{W}) d\lambda \\ &\approx \frac{1}{K} \sum_{k=1}^{K} p(\boldsymbol{x} \mid \boldsymbol{w}, \hat{\lambda}_k) \quad \leftarrow \text{Ensemble} \end{aligned}$$



Marginalize model parameters & architecture

Generative model-based text-to-speech synthesis

- Bayes formulation + factorization + approximations
- Representation: *acoustic features*, *linguistic features*, *mapping*
  - Mapping: Rules $\rightarrow$ HMM $\rightarrow$ NN
  - Feature: Engineered $\rightarrow$ Unsupervised, learned
- Less approximations
  - Joint training, direct waveform modelling
  - Moving towards integrated & Bayesian end-to-end TTS

Naturalness: Concatenative $\leq$ *Generative*

Flexibility: Concatenative $\ll$ *Generative* (e.g., multiple speakers)

Beyond “text”-to-speech synthesis

TTS on conversational assistants

- Texts aren’t fully contained
- Need more context
  - Location to resolve homographs
  - User query to put right emphasis

Beyond “text”-to-speech synthesis

TTS on conversational assistants

- Texts aren’t fully contained
- Need more context
  - Location to resolve homographs
  - User query to put right emphasis

We need representation that can

organize the world information & make it accessible & useful

from TTS generative models

Beyond “generative” TTS

Generative model-based TTS

- Model represents process behind speech production
  - Trained to minimize error against human-produced speech
  - Learned model $\rightarrow$ **speaker**

Beyond “generative” TTS

Generative model-based TTS

- Model represents process behind speech production
  - Trained to minimize error against human-produced speech
  - Learned model $\rightarrow$ **speaker**

- Speech is for communication
  - Goal: maximize the amount of information to be received

Missing “listener”

$\rightarrow$ “listener” in training / model itself?

Thanks!

References I

[1] D. Klatt.
Real-time speech synthesis by rule.
Journal of ASA, 68(S1):S18–S18, 1980.

[2] A. Hunt and A. Black.
Unit selection in a concatenative speech synthesis system using a large speech database.
In *Proc. ICASSP*, pages 373–376, 1996.

[3] K. Tokuda.
Speech synthesis as a statistical machine learning problem.
`https://www.sp.nitech.ac.jp/~tokuda/tokuda_asru2011_for_pdf.pdf`.
Invited talk given at ASRU 2011.

[4] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura.
Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis.
IEICE Trans. Inf. Syst., J83-D-II(11):2099–2107, 2000.
(in Japanese).

[5] H. Zen, K. Tokuda, and A. Black.
Statistical parametric speech synthesis.
Speech Commn., 51(11):1039–1064, 2009.

[6] H. Zen, A. Senior, and M. Schuster.
Statistical parametric speech synthesis using deep neural networks.
In *Proc. ICASSP*, pages 7962–7966, 2013.

[7] Y. Fan, Y. Qian, F.-L. Xie, and F. Soong.
TTS synthesis with bidirectional LSTM based recurrent neural networks.
In *Proc. Interspeech*, pages 1964–1968, 2014.

References II

[8] H. Zen.
Acoustic modeling for speech synthesis: from HMM to RNN.
http://research.google.com/pubs/pub44630.html.
Invited talk given at ASRU 2015.

[9] S. Takaki and J. Yamagishi.
A deep auto-encoder based low-dimensional feature extraction from FFT spectral envelopes for statistical parametric speech synthesis.
In *Proc. ICASSP*, pages 5535–5539, 2016.

[10] G. Hinton, J. McClelland, and D. Rumelhart.
Distributed representation.
In D. Rumelhart, J. McClelland, and the PDP Research Group, editors, *Parallel distributed processing: Explorations in the microstructure of cognition*. MIT Press, 1986.

[11] H. Zen and A. Senior.
Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis.
In *Proc. ICASSP*, pages 3872–3876, 2014.

[12] X. Wang, S. Takaki, and J. Yamagishi.
Investigating very deep highway networks for parametric speech synthesis.
In *Proc. ISCA SSW9*, 2016.

[13] Y. Saito, S. Takamichi, and Saruwatari.
Training algorithm to deceive anti-spoofing verification for DNN-based speech synthesis.
In *Proc. ICASSP*, 2017.

[14] H. Zen, Y. Agiomyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak.
Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices.
In *Proc. Interspeech*, 2016.

References III

[15] P. Muthukumar and A. Black.
A deep learning approach to data-driven parameterizations for statistical parametric speech synthesis.
arXiv:1409.8558, 2014.

[16] P. Wang, Y. Qian, F. Soong, L. He, and H. Zhao.
Word embedding for recurrent neural network based TTS synthesis.
In *Proc. ICASSP*, pages 4879–4883, 2015.

[17] X. Wang, S. Takaki, and J. Yamagishi.
Investigation of using continuous representation of various linguistic units in neural network-based text-to-speech synthesis.
IEICE Trans. Inf. Syst., E90-D(12):2471–2480, 2016.

[18] T. Toda and K. Tokuda.
Statistical approach to vocal tract transfer function estimation based on factor analyzed trajectory hmm.
In *Proc. ICASSP*, pages 3925–3928, 2008.

[19] Y.-J. Wu and K. Tokuda.
Minimum generation error training with direct log spectral distortion on LSPs for HMM-based speech synthesis.
In *Proc. Interspeech*, pages 577–580, 2008.

[20] R. Maia, H. Zen, and M. Gales.
Statistical parametric speech synthesis with joint estimation of acoustic and excitation model parameters.
In *Proc. ISCA SSW7*, pages 88–93, 2010.

[21] K. Tokuda and H. Zen.
Directly modeling speech waveforms by neural networks for statistical parametric speech synthesis.
In *Proc. ICASSP*, pages 4215–4219, 2015.

[22] K. Tokuda and H. Zen.
Directly modeling voiced and unvoiced components in speech waveforms by neural networks.
In *Proc. ICASSP*, pages 5640–5644, 2016.

References IV

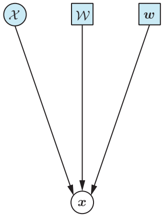
[23] F. Itakura and S. Saito.
A statistical method for estimation of speech spectral density and formant frequencies.
Trans. IEICE, J53A:35–42, 1970.

[24] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu.
WaveNet: A generative model for raw audio.
arXiv:1609.03499, 2016.

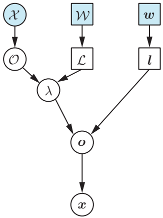
[25] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio.
SampleRNN: An unconditional end-to-end neural audio generation model.
arXiv:1612.07837, 2016.

[26] S. Imai and C. Furuichi.
Unbiased estimation of log spectrum.
In *Proc. EURASIP*, pages 203–206, 1988.

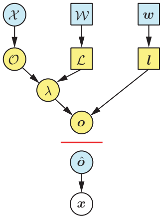
[27] H. Kameoka, Y. Ohishi, D. Mochihashi, and J. Le Roux.
Speech analysis with multi-kernel linear prediction.
In *Proc. Spring Conference of ASJ*, pages 499–502, 2010.
(in Japanese).



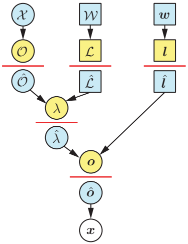
(1) Bayesian



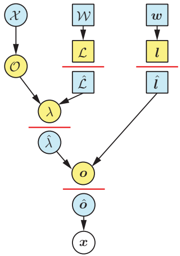
(2) Auxiliary variables
+ factorization



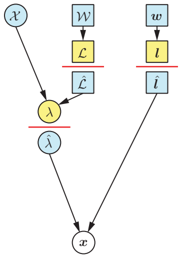
(3) Joint maximization



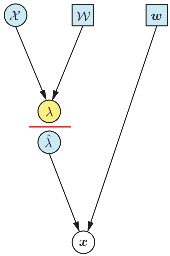
(4) Step-by-step maximization
e.g., statistical parametric TTS



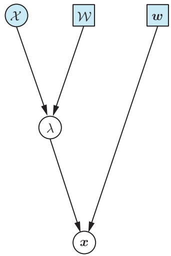
(5) Joint acoustic feature
extraction + model training



(6) Conditional WaveNet
-based TTS



(7) Integrated end-to-end



(8) Bayesian end-to-end