Statistical Parametric Speech Synthesis: From HMM to LSTM-RNN

Heiga Zen
Google
July 29th, 2015

Outline

Basics of HMM-based speech synthesis
- Background
- HMM-based speech synthesis

Advanced topics in HMM-based speech synthesis
- Flexibility
- Improve naturalness

Neural network-based speech synthesis
- Feed-forward neural network (DNN & DMDN)
- Recurrent neural network (RNN & LSTM-RNN)
- Results

Conclusion

Lecturer

- Heiga Zen
- PhD from Nagoya Institute of Technology, Japan (2006)
- Intern, IBM T.J. Watson Research, New York (2004–2005)
- Research engineer, Toshiba Research Europe, Cambridge (2009–2011)
- Research scientist, Google, London (2011–Present)

Outline

Basics of HMM-based speech synthesis

- Background
- HMM-based speech synthesis

Advanced topics in HMM-based speech synthesis

- Flexibility
- Improve naturalness

Neural network-based speech synthesis

- Feed-forward neural network (DNN & DMDN)
- Recurrent neural network (RNN & LSTM-RNN)
- Results

Conclusion

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

Automatic speech recognition (ASR)

Speech (real-valued time series) $\rightarrow$ Text (discrete symbol sequence)

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

Automatic speech recognition (ASR)

Speech (real-valued time series) $\rightarrow$ Text (discrete symbol sequence)

Statistical machine translation (SMT)

Text (discrete symbol sequence) $\rightarrow$ Text (discrete symbol sequence)

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

Automatic speech recognition (ASR)

Speech (real-valued time series) $\longrightarrow$ Text (discrete symbol sequence)

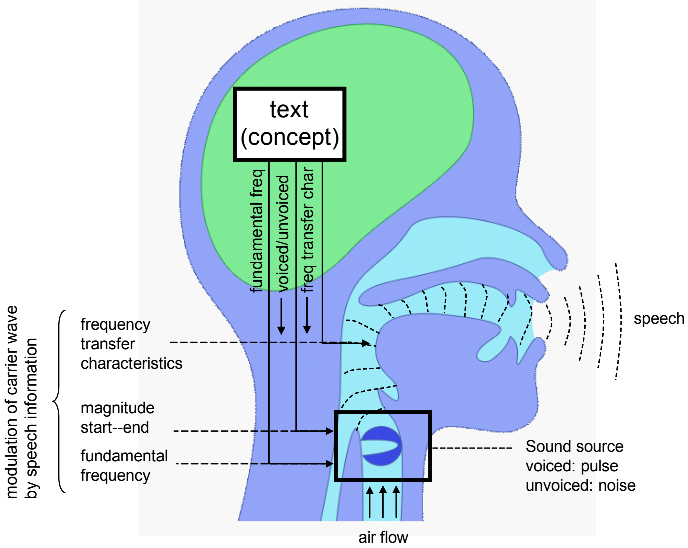
Statistical machine translation (SMT)

Text (discrete symbol sequence) $\longrightarrow$ Text (discrete symbol sequence)

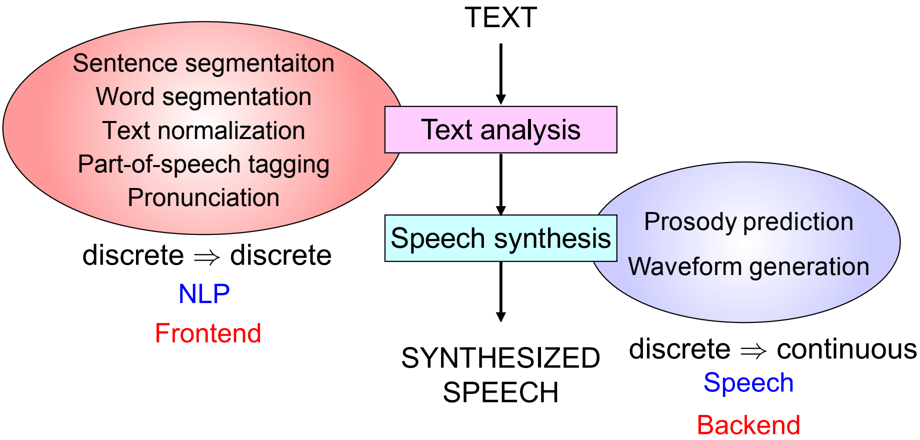
Text-to-speech synthesis (TTS)

Text (discrete symbol sequence) $\longrightarrow$ Speech (real-valued time series)

Speech production process

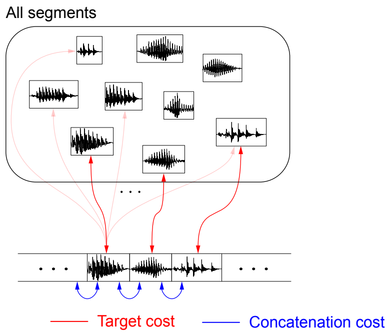


Typical flow of TTS system



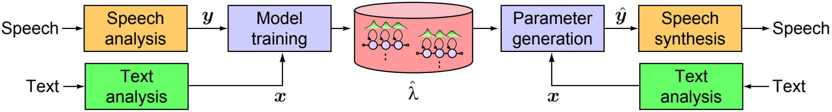
This presentation mainly talks about backend

Concatenative, unit selection speech synthesis



- Concatenate actual instances of speech from database
- Large data + automatic learning
  → High-quality synthetic voices can be built automatically
- Single inventory per unit → diphone synthesis [1]
- Multiple inventory per unit → unit selection synthesis [2]

Statistical parametric speech synthesis (SPSS) [3]

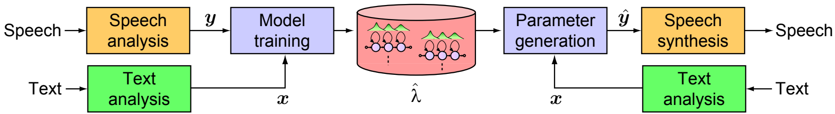


Training

- Extract linguistic features $\boldsymbol{x}$ & acoustic features $\boldsymbol{y}$
- Train acoustic model λ given $(\boldsymbol{x}, \boldsymbol{y})$

$$\hat{\lambda} = \arg\max p(\boldsymbol{y} \mid \boldsymbol{x}, \lambda)$$

Statistical parametric speech synthesis (SPSS) [3]



Training

- Extract linguistic features $\boldsymbol{x}$ & acoustic features $\boldsymbol{y}$
- Train acoustic model λ given $(\boldsymbol{x}, \boldsymbol{y})$

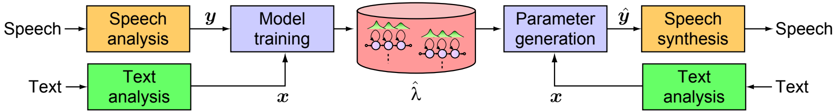
$$\hat{\lambda} = \arg\max p(\boldsymbol{y} \mid \boldsymbol{x}, \lambda)$$

Synthesis

- Extract $\boldsymbol{x}$ from text to be synthesized
- Generate most probable $\boldsymbol{y}$ from $\hat{\lambda}$ then reconstruct waveform

$$\hat{\boldsymbol{y}} = \arg\max p(\boldsymbol{y} \mid \boldsymbol{x}, \hat{\lambda})$$

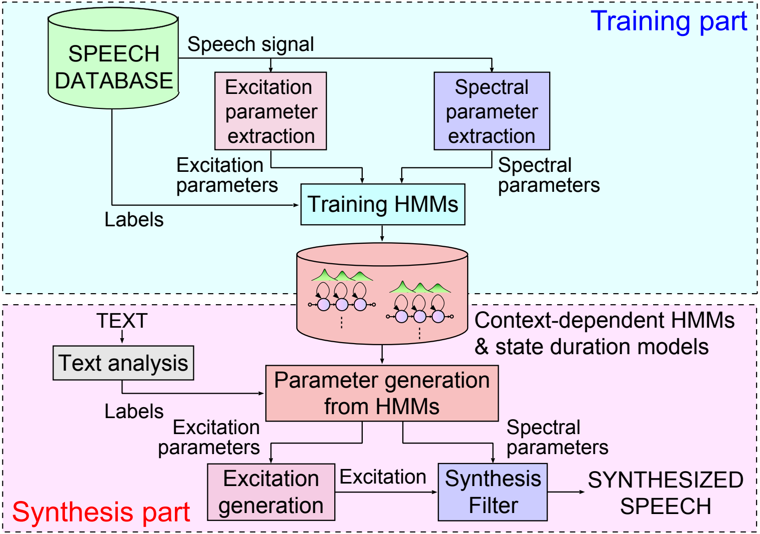
Statistical parametric speech synthesis (SPSS) [3]



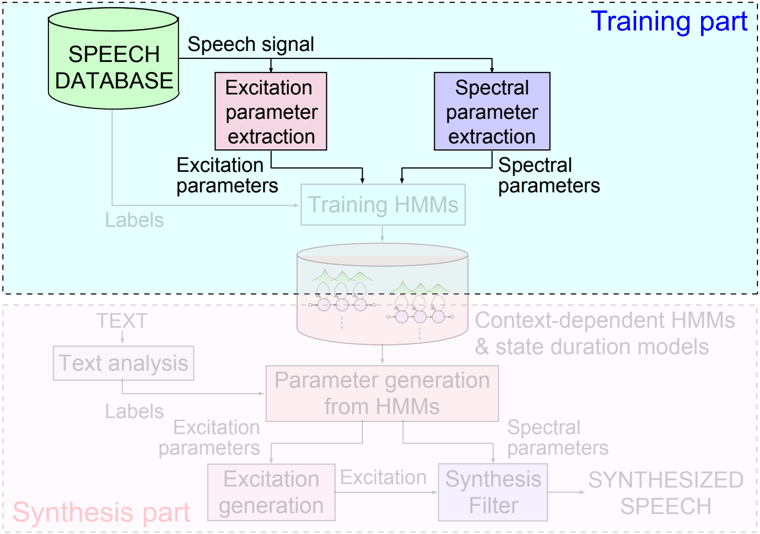
- Vocoded speech (buzzy or muffled)
- Small footprint

Hidden Markov model (HMM) as its acoustic model
→ HMM-based speech synthesis system (HTS) [4]

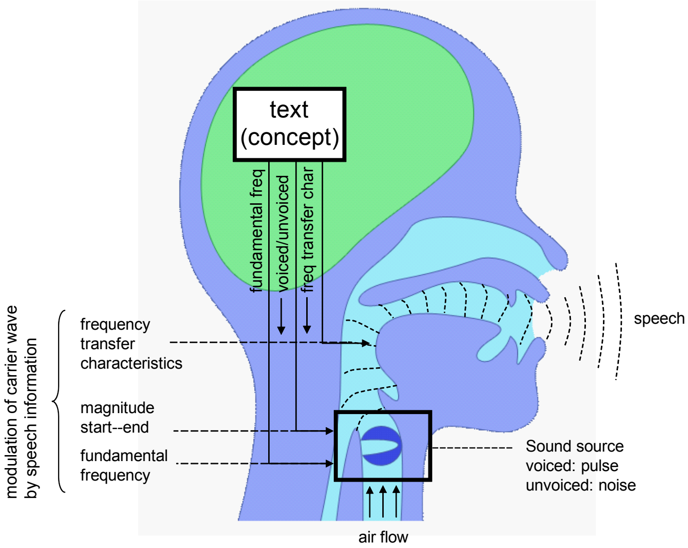
HMM-based speech synthesis [4]



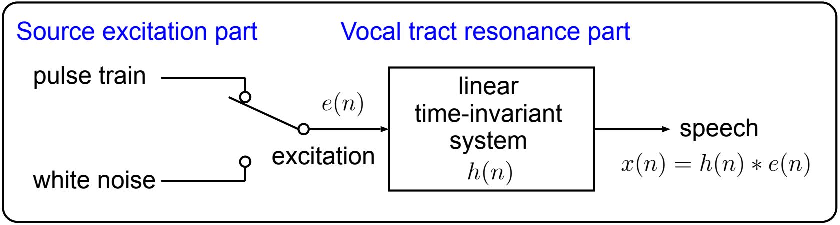
HMM-based speech synthesis [4]



Speech production process



Source-filter model



$$x(n) = h(n) * e(n)$$

$\downarrow$ Fourier transform

$$X(e^{j\omega}) = H(e^{j\omega})E(e^{j\omega})$$

$H\left(e^{j\omega}\right)$ should be defined by HMM state-output vectors
e.g., mel-cepstrum, line spectral pairs

Parametric models of speech signal

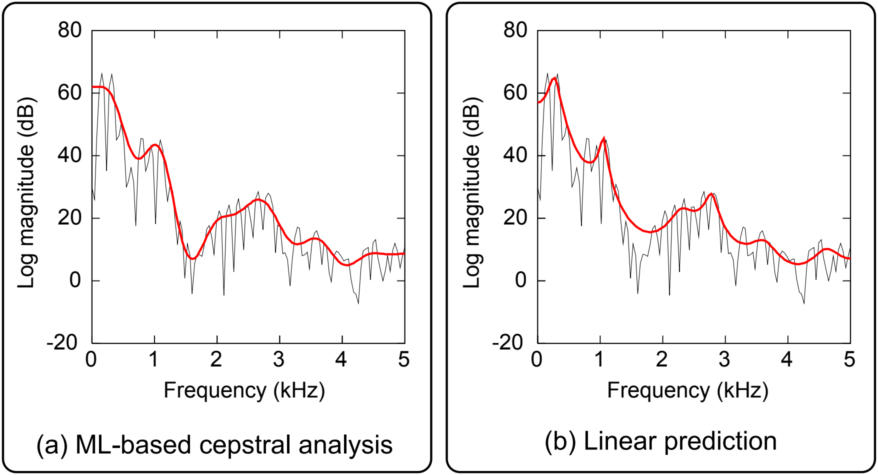
Autoregressive (AR) model	Exponential (EX) model
$H(z) = \dfrac{K}{1 - \sum_{m=0}^{M} c(m) z^{-m}}$	$H(z) = \exp \sum_{m=0}^{M} c(m) z^{-m}$

Estimate model parameters based on ML

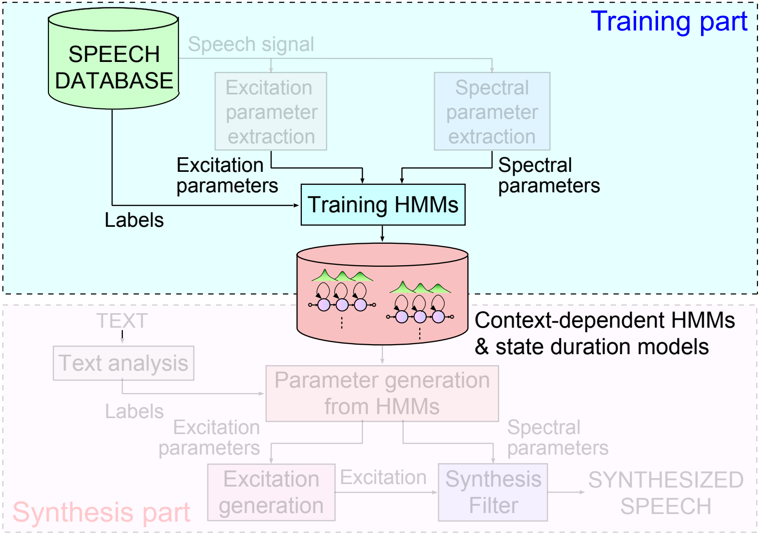
$$\boldsymbol{c} = \arg\max_{\boldsymbol{c}} p(\boldsymbol{x} \mid \boldsymbol{c})$$

- $p(\boldsymbol{x} \mid \boldsymbol{c})$: AR model $\rightarrow$ Linear predictive analysis [5]
- $p(\boldsymbol{x} \mid \boldsymbol{c})$: EX model $\rightarrow$ (ML-based) cepstral analysis [6]

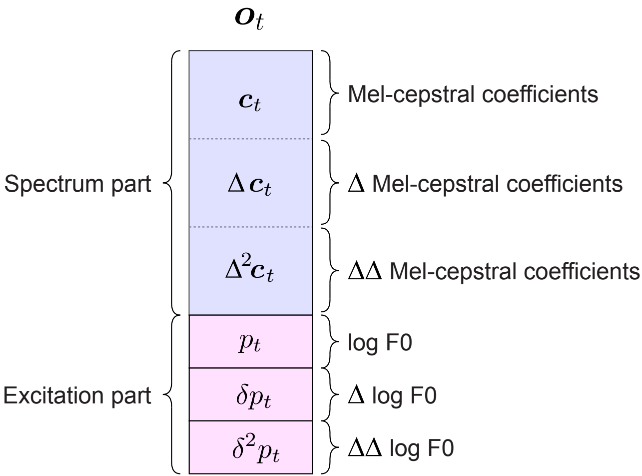
Examples of speech spectra



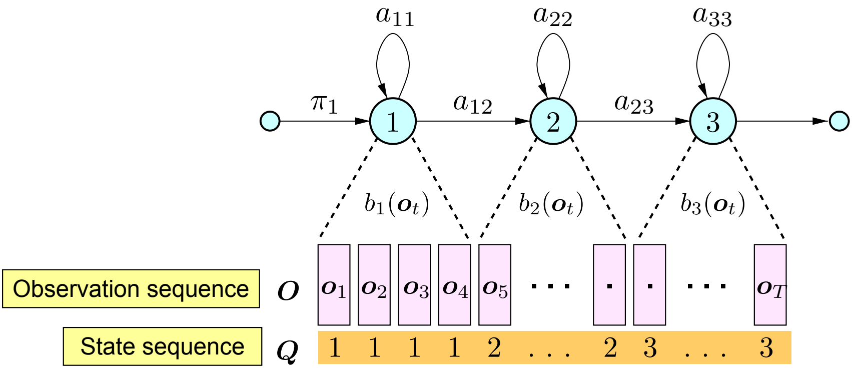
HMM-based speech synthesis [4]



Structure of state-output (observation) vectors

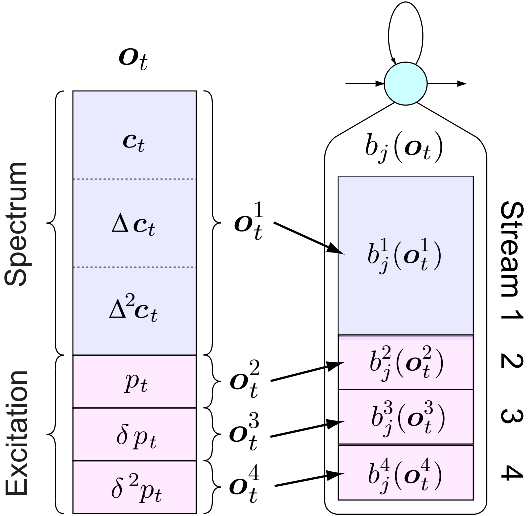


Hidden Markov model (HMM)

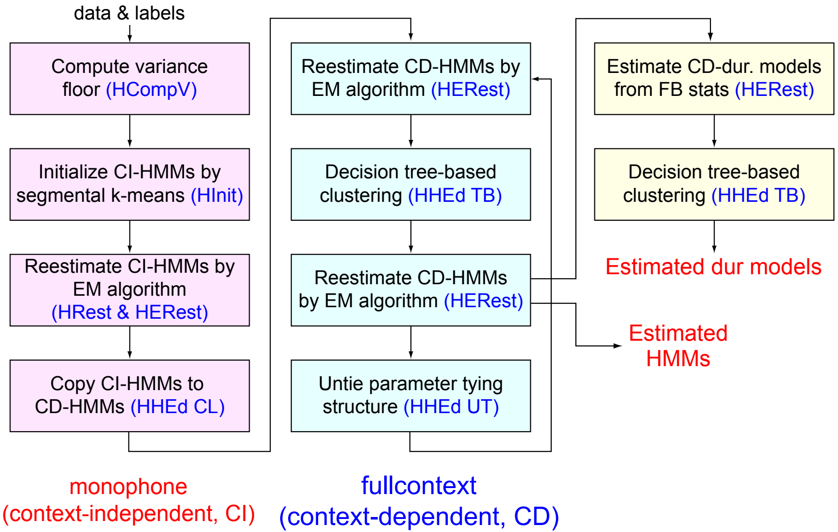


Multi-stream HMM structure

$$b_j(\boldsymbol{o}_t) = \prod_{s=1}^{S} \left(b_j^s(\boldsymbol{o}_t^s) \right)^{w_s}$$



Training process



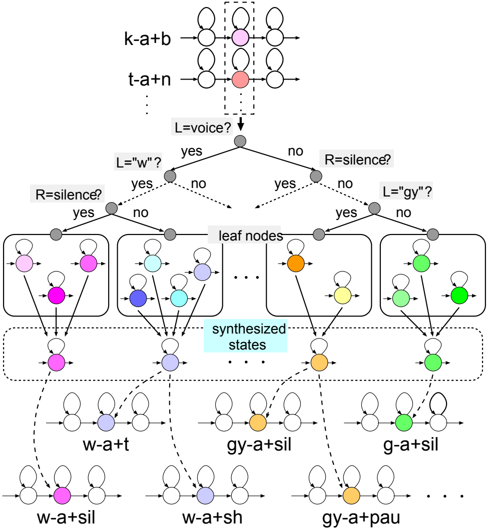
Context-dependent acoustic modeling [4]

- {preceding, succeeding} two phonemes
- Position of current phoneme in current syllable
- # of phonemes at {preceding, current, succeeding} syllable
- {accent, stress} of {preceding, current, succeeding} syllable
- Position of current syllable in current word
- # of {preceding, succeeding} {stressed, accented} syllables in phrase
- # of syllables {from previous, to next} {stressed, accented} syllable
- Guess at part of speech of {preceding, current, succeeding} word
- # of syllables in {preceding, current, succeeding} word
- Position of current word in current phrase
- # of {preceding, succeeding} content words in current phrase
- # of words {from previous, to next} content word
- # of syllables in {preceding, current, succeeding} phrase

...

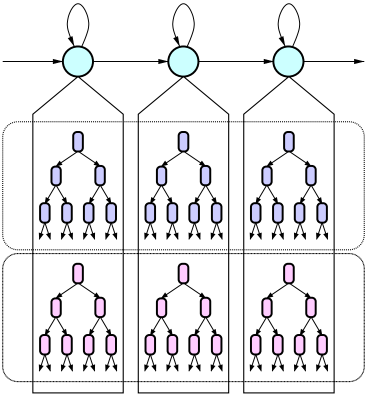
Impossible to have all possible models

Decision tree-based state clustering [7]



Stream-dependent tree-based clustering

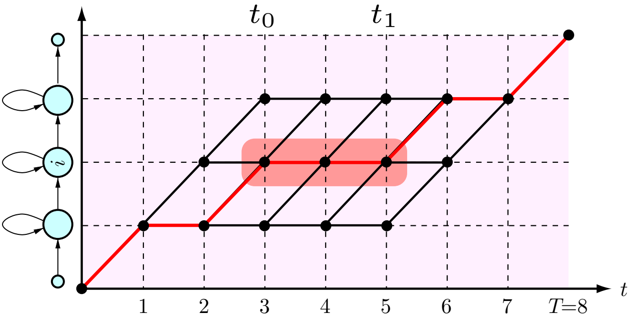
Decision trees
for
mel-cepstrum



Decision trees
for F0

Spectrum & excitation can have different context dependency
$\rightarrow$ Build decision trees individually

State duration models [8]

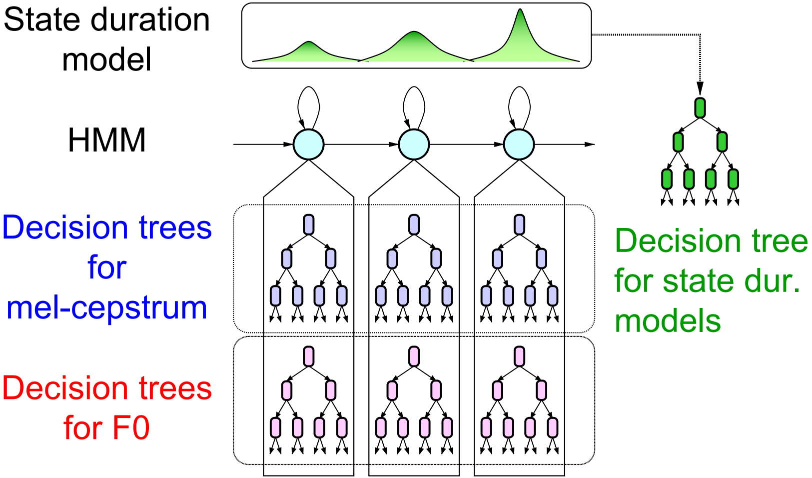


Probability to enter state i at t_0 then leave at $t_1 + 1$

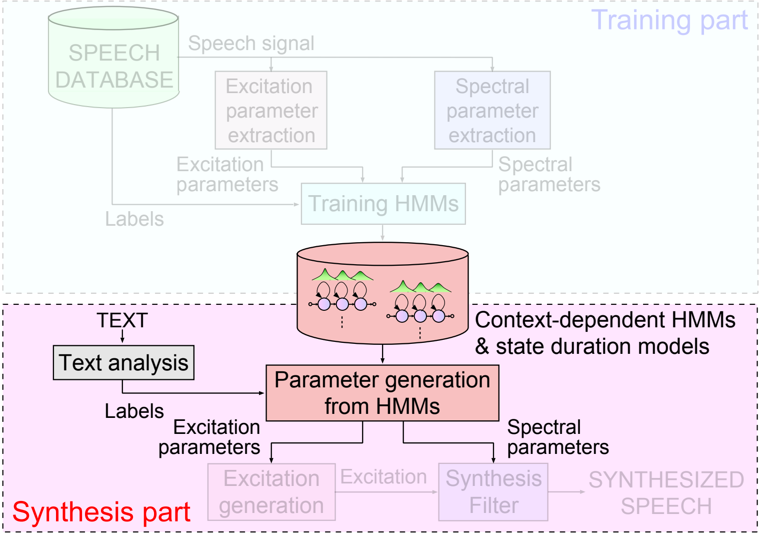
$$\chi_{t_0,t_1}(i) \propto \sum_{j \neq i} \alpha_{t_0-1}(j) a_{ji} a_{ii}^{t_1-t_0} \prod_{t=t_0}^{t_1} b_i(\boldsymbol{o}_t) \sum_{k \neq i} a_{ik} b_k(\boldsymbol{o}_{t_1+1}) \beta_{t_1+1}(k)$$

$\rightarrow$ estimate state duration models

Stream-dependent tree-based clustering



HMM-based speech synthesis [4]



Speech parameter generation algorithm [9]

Generate most probable state outputs given HMM and words

$$
\begin{aligned}
\hat{\boldsymbol{o}} &= \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid w, \hat{\lambda}) \\
&= \arg\max_{\boldsymbol{o}} \sum_{\forall \boldsymbol{q}} p(\boldsymbol{o}, \boldsymbol{q} \mid w, \hat{\lambda}) \\
&\approx \arg\max_{\boldsymbol{o}} \max_{\boldsymbol{q}} p(\boldsymbol{o}, \boldsymbol{q} \mid w, \hat{\lambda}) \\
&= \arg\max_{\boldsymbol{o}} \max_{\boldsymbol{q}} p(\boldsymbol{o} \mid \boldsymbol{q}, \hat{\lambda}) P(\boldsymbol{q} \mid w, \hat{\lambda})
\end{aligned}
$$

Speech parameter generation algorithm [9]

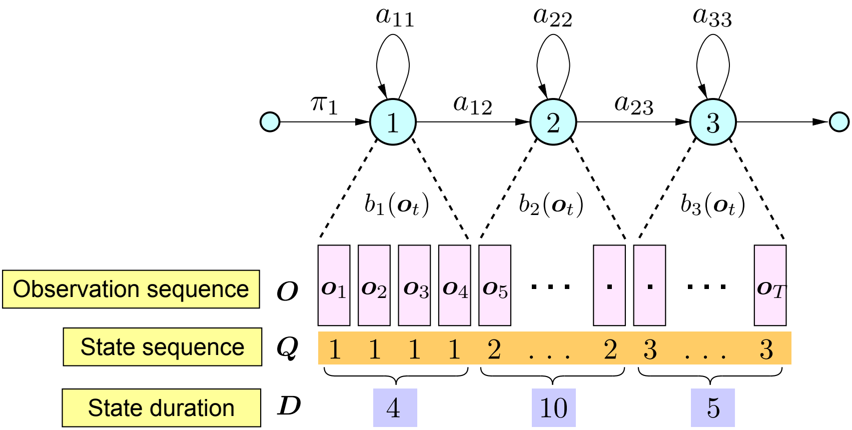
Generate most probable state outputs given HMM and words

$$
\begin{aligned}
\hat{\boldsymbol{o}} &= \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid w, \hat{\lambda}) \\
&= \arg\max_{\boldsymbol{o}} \sum_{\forall \boldsymbol{q}} p(\boldsymbol{o}, \boldsymbol{q} \mid w, \hat{\lambda}) \\
&\approx \arg\max_{\boldsymbol{o}} \max_{\boldsymbol{q}} p(\boldsymbol{o}, \boldsymbol{q} \mid w, \hat{\lambda}) \\
&= \arg\max_{\boldsymbol{o}} \max_{\boldsymbol{q}} p(\boldsymbol{o} \mid \boldsymbol{q}, \hat{\lambda}) P(\boldsymbol{q} \mid w, \hat{\lambda})
\end{aligned}
$$

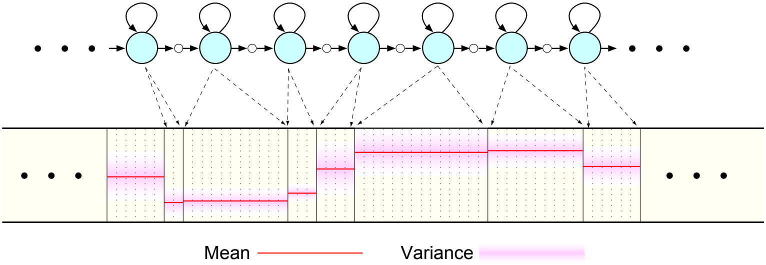
Determine the best state sequence and outputs sequentially

$$
\begin{aligned}
\hat{\boldsymbol{q}} &= \arg\max_{\boldsymbol{q}} P(\boldsymbol{q} \mid w, \hat{\lambda}) \\
\hat{\boldsymbol{o}} &= \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{q}}, \hat{\lambda})
\end{aligned}
$$

Best state sequence



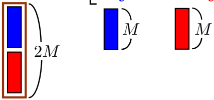
Best state outputs
w/o dynamic features

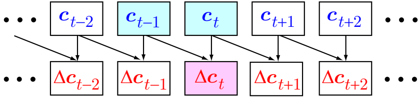


$\hat{o}$ becomes step-wise mean vector sequence

State output vectors include static & dynamic features

$$\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta \boldsymbol{c}_t^\top\right]^\top \qquad \Delta \boldsymbol{c}_t = \boldsymbol{c}_t - \boldsymbol{c}_{t-1}$$





Relationship between static and dynamic features can be arranged as

$$\underbrace{\begin{array}{c} \\ \boldsymbol{o}_{t-1} \\ \\ \boldsymbol{o}_{t} \\ \\ \boldsymbol{o}_{t+1} \\ \\ \end{array}\begin{bmatrix} \vdots \\ \boldsymbol{c}_{t-1} \\ \Delta\boldsymbol{c}_{t-1} \\ \boldsymbol{c}_{t} \\ \Delta\boldsymbol{c}_{t} \\ \boldsymbol{c}_{t+1} \\ \Delta\boldsymbol{c}_{t+1} \\ \vdots \end{bmatrix}}_{\boldsymbol{o}} = \underbrace{\begin{bmatrix} \cdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ \cdots & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \cdots \\ \cdots & -\boldsymbol{I} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \cdots \\ \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \cdots \\ \cdots & \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{I} & \boldsymbol{0} & \cdots \\ \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \cdots \\ \cdots & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{I} & \cdots \\ \cdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{bmatrix}}_{\boldsymbol{W}} \underbrace{\begin{bmatrix} \vdots \\ \boldsymbol{c}_{t-2} \\ \boldsymbol{c}_{t-1} \\ \boldsymbol{c}_{t} \\ \boldsymbol{c}_{t+1} \\ \vdots \end{bmatrix}}_{\boldsymbol{c}}$$

Speech parameter generation algorithm [9]

Introduce dynamic feature constraints

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{q}}, \hat{\lambda}) \quad \text{subject to} \quad \boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$$

Speech parameter generation algorithm [9]

Introduce dynamic feature constraints

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{q}}, \hat{\lambda}) \quad \text{subject to} \quad \boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$$

If state-output distribution is single Gaussian

$$p(\boldsymbol{o} \mid \hat{\boldsymbol{q}}, \hat{\lambda}) = \mathcal{N}(\boldsymbol{o}; \hat{\boldsymbol{\mu}}_{\hat{\boldsymbol{q}}}, \hat{\boldsymbol{\Sigma}}_{\hat{\boldsymbol{q}}})$$

Speech parameter generation algorithm [9]

Introduce dynamic feature constraints

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{q}}, \hat{\lambda}) \quad \text{subject to} \quad \boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$$

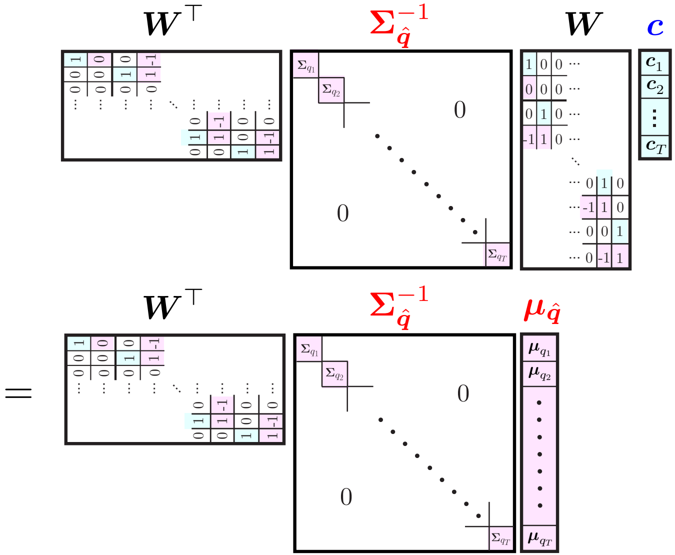
If state-output distribution is single Gaussian

$$p(\boldsymbol{o} \mid \hat{\boldsymbol{q}}, \hat{\lambda}) = \mathcal{N}(\boldsymbol{o}; \hat{\boldsymbol{\mu}}_{\hat{\boldsymbol{q}}}, \hat{\boldsymbol{\Sigma}}_{\hat{\boldsymbol{q}}})$$

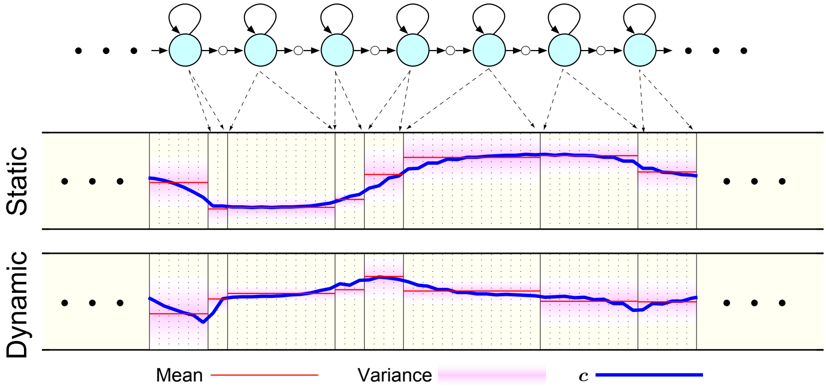
By setting $\partial \log \mathcal{N}(\boldsymbol{W}\boldsymbol{c}; \hat{\boldsymbol{\mu}}_{\hat{\boldsymbol{q}}}, \hat{\boldsymbol{\Sigma}}_{\hat{\boldsymbol{q}}}) / \partial \boldsymbol{c} = \boldsymbol{0}$

$$\boldsymbol{W}^\top \hat{\boldsymbol{\Sigma}}_{\hat{\boldsymbol{q}}}^{-1} \boldsymbol{W}\boldsymbol{c} = \boldsymbol{W}^\top \hat{\boldsymbol{\Sigma}}_{\hat{\boldsymbol{q}}}^{-1} \hat{\boldsymbol{\mu}}_{\hat{\boldsymbol{q}}}$$

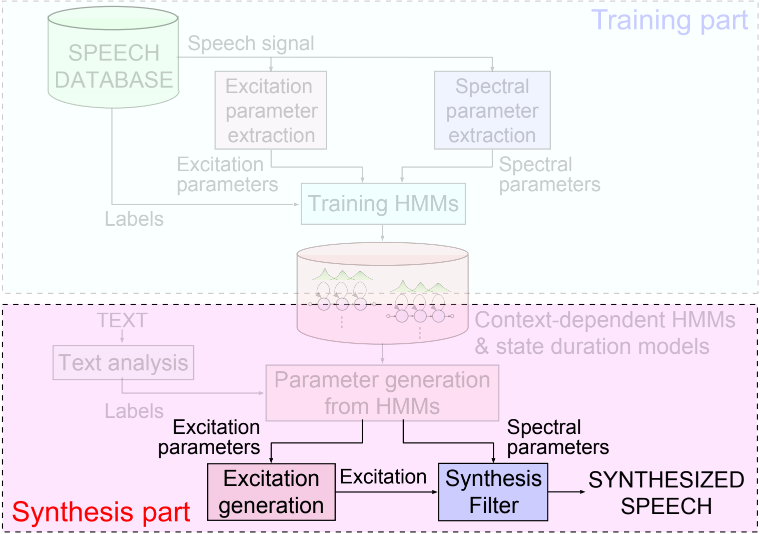
Speech parameter generation algorithm [9]



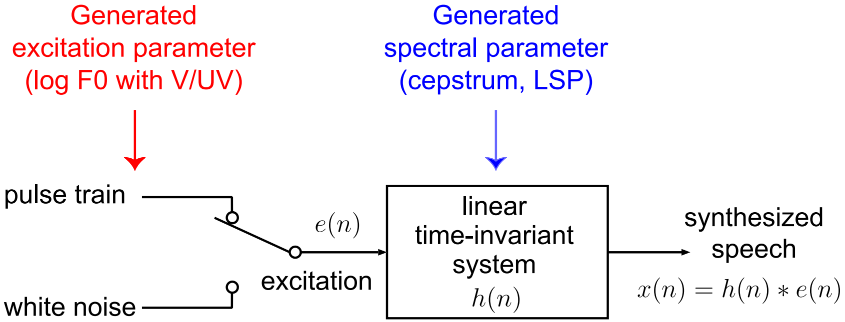
Generated speech parameter trajectory



HMM-based speech synthesis [4]



Waveform reconstruction



Synthesis filter

- Cepstrum $\longrightarrow$ LMA filter
- Generalized cepstrum $\longrightarrow$ GLSA filter
- Mel-cepstrum $\longrightarrow$ MLSA filter
- Mel-generalized cepstrum $\longrightarrow$ MGLSA filter
- LSP $\longrightarrow$ LSP filter
- PARCOR $\longrightarrow$ all-pole lattice filter
- LPC $\longrightarrow$ all-pole filter

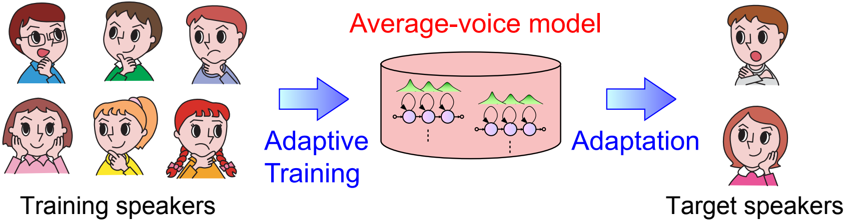
Any questions?

Outline

Advantages

- Flexibility to change voice characteristics
- Small footprint
- More data

Adaptation (mimicking voice) [10]



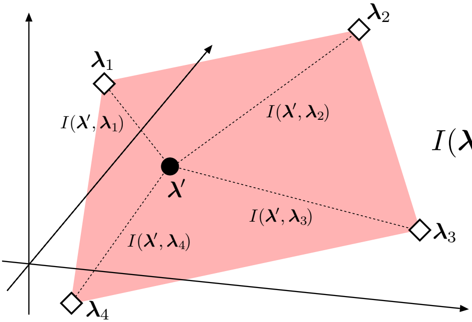
- Train average voice model (AVM) from training speakers using SAT
- Adapt AVM to target speakers
- Requires small data from target speaker/speaking style
  → Small cost to create new voices

Adaptation demo

- **Speaker adaptation**
  - VIP voice: GWB   BHO
  - Child voice:
- **Style adaptation (in Japanese)**
  - Joyful
  - Sad
  - Rough

From http://homepages.inf.ed.ac.uk/jyamagis/Demo-html/demo.html

Interpolation (mixing voice) [11, 12, 13, 14]



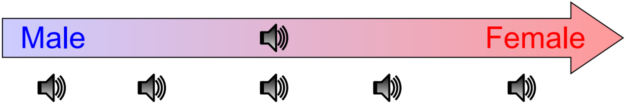
$\boldsymbol{\lambda}$: HMM set

$I(\boldsymbol{\lambda}', \boldsymbol{\lambda})$: Interpolation ratio

- Interpolate representive HMM sets
- Can obtain new voices w/o adaptation data
- Eigenvoice / CAT / multiple regression
  $\longrightarrow$ estimate representative HMM sets from data

Interpolation demo (1)

- **Speaker interpolation (in Japanese)**
  - Male & Female

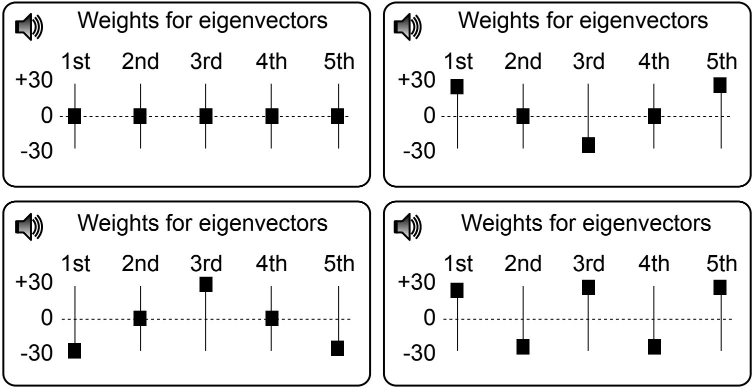


- **Style interpolation**
  - Neutral → Angry
  - Neutral → Happy

From http://www.sp.nitech.ac.jp/
& http://homepages.inf.ed.ac.uk/jyamagis/Demo-html/demo.html

Interpolation demo (2)

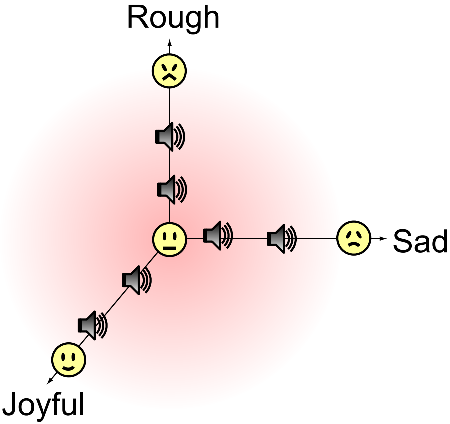
Speaker characteristics modification



From http://www.sp.nitech.ac.jp/~demo/synthesis_demo_2001/

Interpolation demo (3)

Style-control



From http://homepages.inf.ed.ac.uk/jyamagis/Demo-html/demo.html

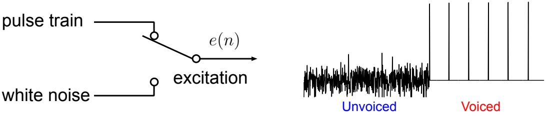
Drawbacks

- **Quality**
  buzzy, muffled synthetic speech

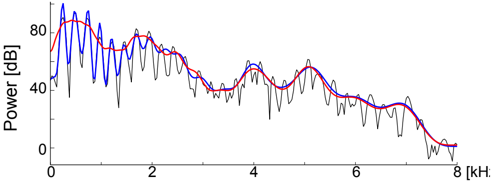
- **Major factors for quality degradation [3]**
  - Vocoder (speech analysis & synthesis)
  - Acoustic model (HMM)
  - Oversmoothing (parameter generation)

Vocoding issues

- **Simple pulse / noise excitation**
  Difficult to model mix of V/UV sounds (e.g., voiced fricatives)



- **Spectral envelope extraction**
  Harmonic effect often cause problem

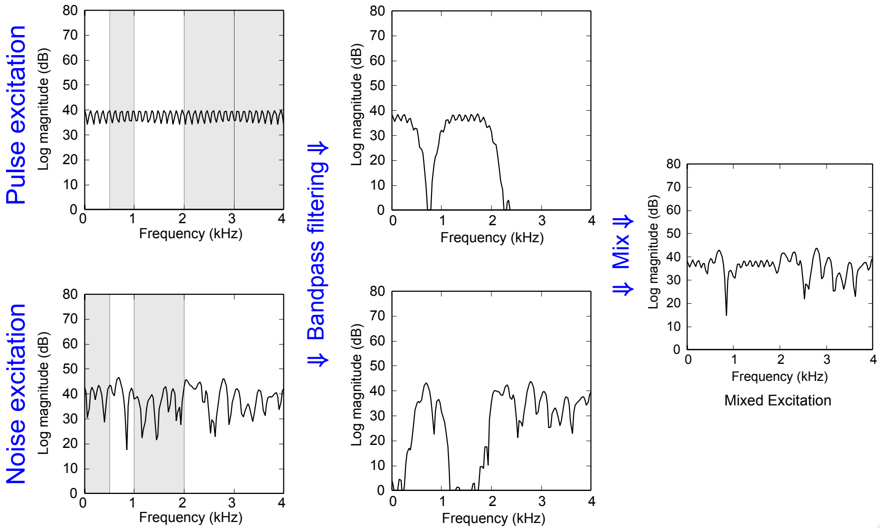


- **Phase**
  Important but usually ignored

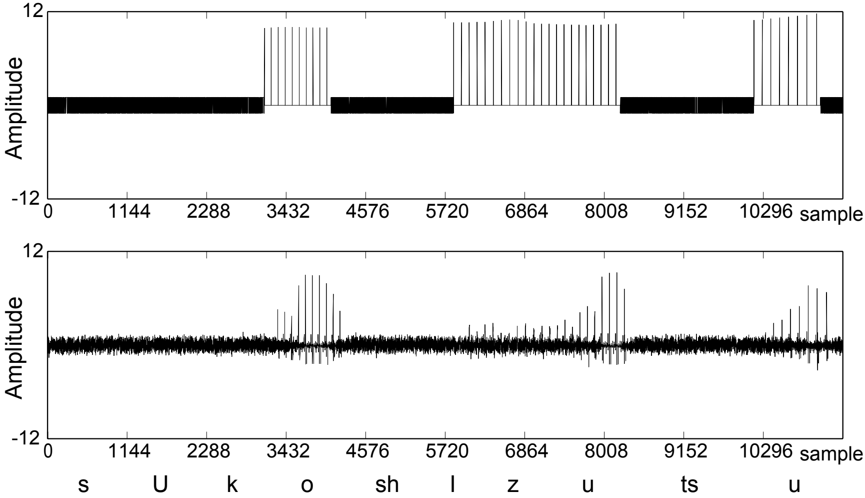
Better vocoding

- **Mixed excitation linear prediction (MELP)**
- **STRAIGHT**
- Multi-band excitation
- Harmonic + noise model (HNM)
- Harmonic / stochastic model
- LF model
- Glottal waveform
- Residual codebook
- **ML excitation**

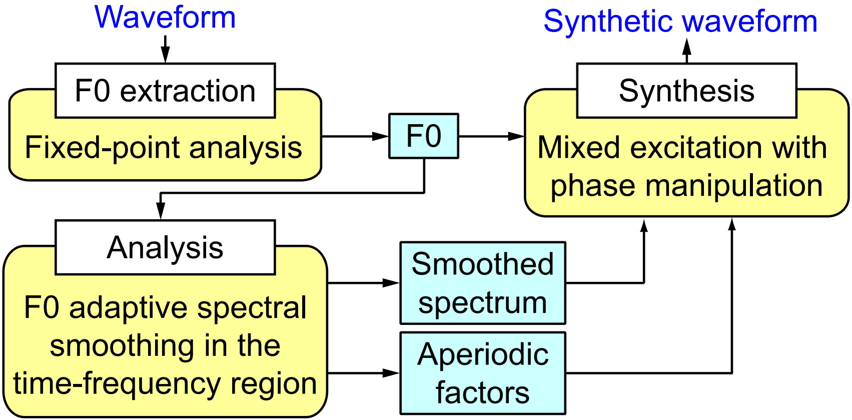
MELP-style mixed excitation [15]



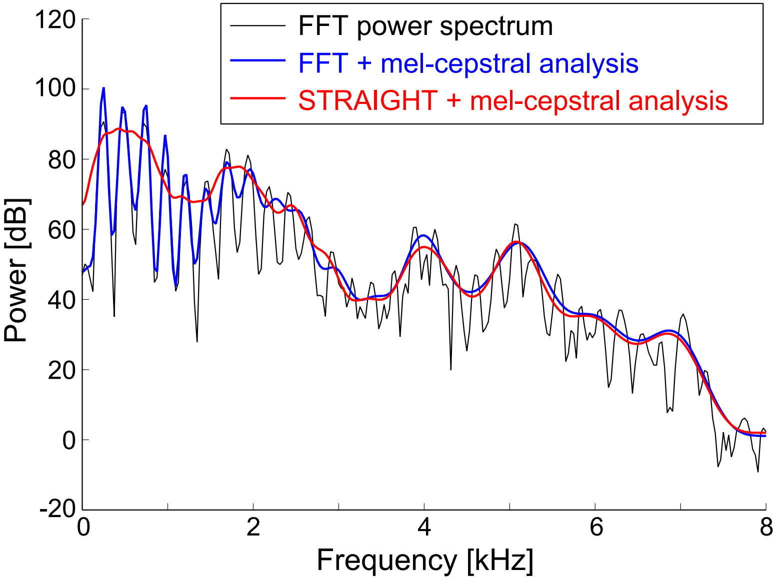
MELP-style mixed excitation [15]



STRAIGHT [16]

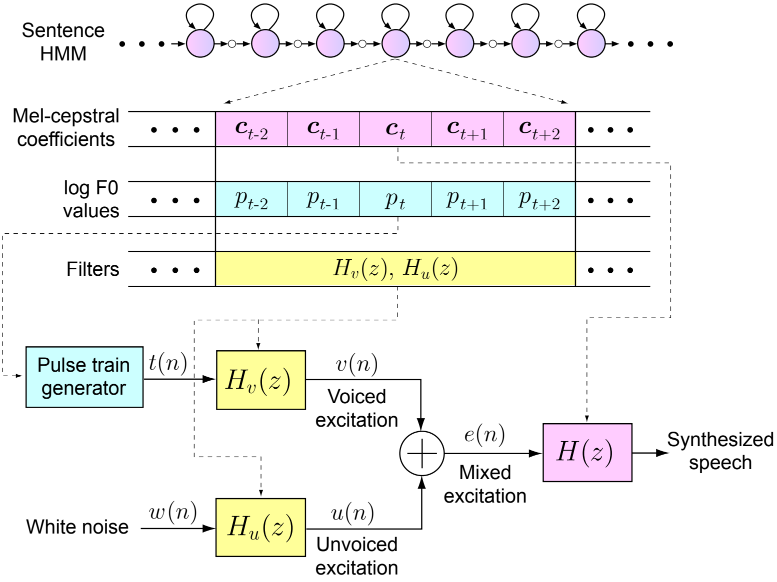

Waveform
Synthetic waveform
F0 extraction
Fixed-point analysis
F0
Synthesis
Mixed excitation with phase manipulation
Analysis
F0 adaptive spectral smoothing in the time-frequency region
Smoothed spectrum
Aperiodic factors

STRAIGHT [16]

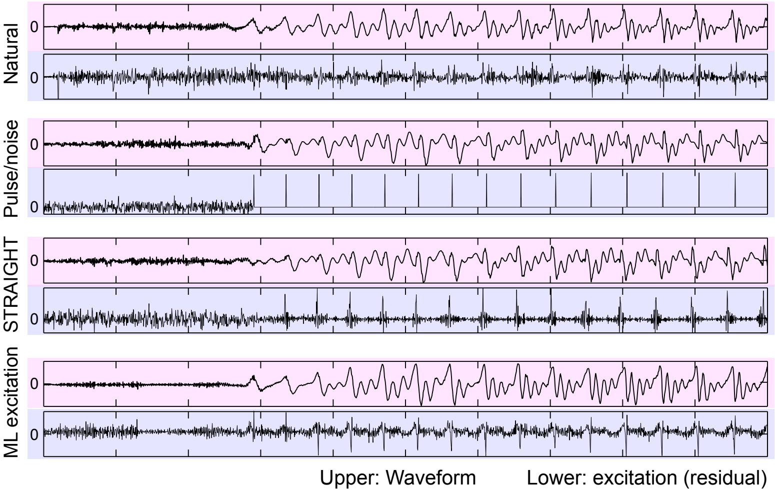


FFT power spectrum
FFT + mel-cepstral analysis
STRAIGHT + mel-cepstral analysis
Power [dB]
120
100
80
60
40
20
0
-20
0
2
4
6
8
Frequency [kHz]

Trainable excitation model [17]



Trainable excitation model [17]



Limitations of HMMs for acoustic modeling

- **Piece-wise constatnt statistics**
  Statistics do not vary within an HMM state
- **Conditional independence assumption**
  State output probability depends only on the current state
- **Weak duration modeling**
  State duration probability decreases exponentially with time

None of them hold for real speech

Better acoustic modeling

- **Piece-wise constatnt statistics** → Dynamical model
  - Trended HMM, autoregressive HMM (ARHMM)
  - Polynomial segment model, hidden trajectory model (HTM)
  - **Trajectory HMM**
- **Conditional independence assumption** → Graphical model
  - Buried Markov model, ARHMM, linear dynamical model (LDM)
  - HTM, Gaussian process (GP)
  - Trajectory HMM
- **Weak duration modeling** → Explicit duration model
  - Hidden semi-Markov model

Trajectory HMM [18]

- Derived from HMM by imposing dynamic feature constraints
- Underlying generative model in HMM-based speech synthesis

$$p(\boldsymbol{c} \mid \lambda) = \sum_{\forall \boldsymbol{q}} p(\boldsymbol{c} \mid \boldsymbol{q}, \lambda) P(\boldsymbol{q} \mid \lambda)$$

$$p(\boldsymbol{c} \mid \boldsymbol{q}, \lambda) = \mathcal{N}\left(\boldsymbol{c}; \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right)$$

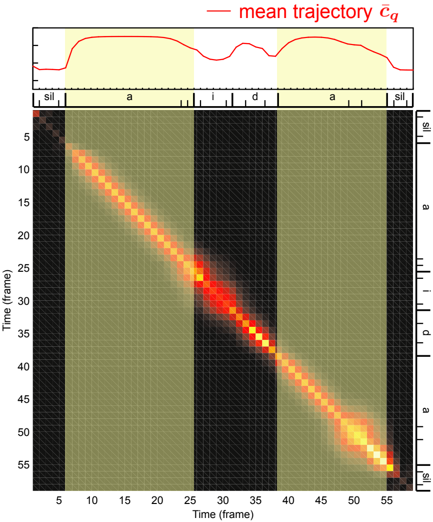
where

$$\boldsymbol{P}_{\boldsymbol{q}}^{-1} = \boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{W}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W}$$

$$\boldsymbol{r}_{\boldsymbol{q}} = \boldsymbol{W}^{\top} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}}$$

$$\bar{\boldsymbol{c}}_{\boldsymbol{q}} = \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{r}_{\boldsymbol{q}}$$

Trajectory HMM [18]



Temporal covariance matrix P_q

Relation to HMM-based speech synthesis

- Mean vector of trajectory HMM

$$\boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W} \bar{\boldsymbol{c}}_{\boldsymbol{q}} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}}$$

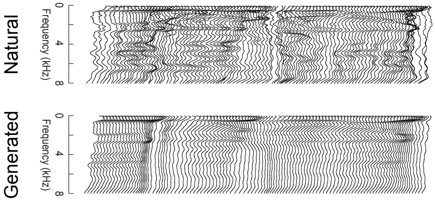
- Speech parameter trajectory used in HMM-based speech synthesis

$$\boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W} \boldsymbol{c} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}}$$

ML estimation of trajectory HMM
$\rightarrow$ Make training & synthesis consistent

Oversmoothing

- **Speech parameter generation algorithm**
  - Dynamic feature constraints make generated parameters smooth
  - Often too smooth → sounds muffled

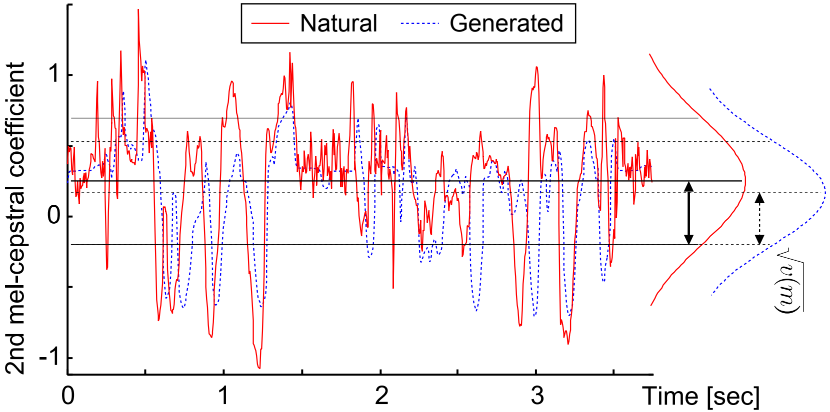


- **Why?**
  - Details of spectral (formant) structure disappear
  - Use of better AM relaxes the issue, but not enough

Oversmoothing compensation

- **Postfiltering**
  - Mel-cepstrum
  - LSP
- **Nonparametric approach**
  - Conditional parameter generation
  - Discrete HMM-based speech synthesis
- **Combine multiple-level statistics**
  - **Global variance (intra-utterance variance)**
  - Modulation spectrum (intra-utterance frequency components)

Global variance [19]



GVs of synthesized speech are typically narrower

Speech parameter generation with GV [19]

- **Speech parameter generation**

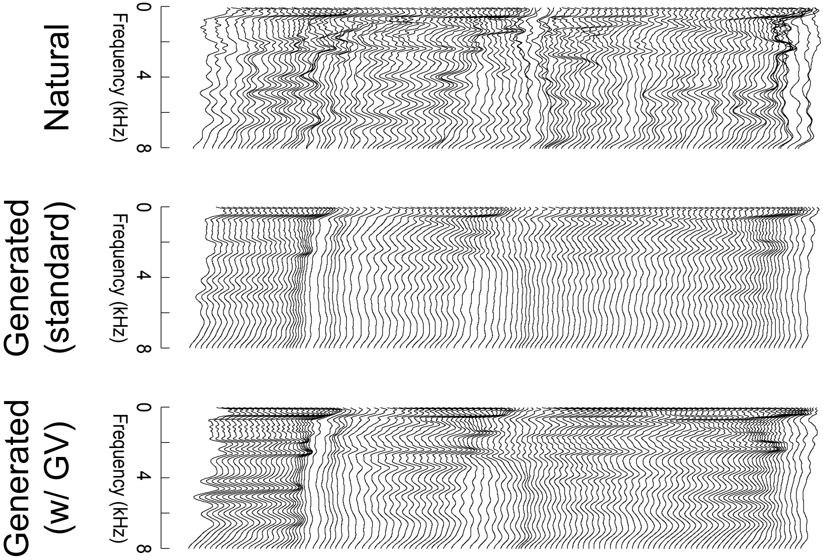
$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \; \log \mathcal{N}\left(\boldsymbol{W}\boldsymbol{c}; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right)$$

- **Speech parameter generation w/ GV**

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \; \log \mathcal{N}\left(\boldsymbol{W}\boldsymbol{c}; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right) + \omega \log \mathcal{N}\left(v(\boldsymbol{c}); \boldsymbol{\mu}_{v}, \boldsymbol{\Sigma}_{v}\right)$$

2nd term works as a penalty for oversmoothing

Effect of GV



Natural
Generated (standard)
Generated (w/ GV)
Frequency (kHz)
0
4
8

Any questions?

Outline

Characteristics of SPSS

- **Advantages**
  - Flexibility to change voice characteristics
    - Adaptation
    - Interpolation / eigenvoice / CAT / multiple regression
  - Small footprint
  - Robustness

- **Drawback**
  - Quality

- **Major factors for quality degradation [3]**
  - Vocoder (speech analysis & synthesis)
  - Acoustic model (HMM) → Neural networks
  - Oversmoothing (parameter generation)

Linguistic $\rightarrow$ acoustic mapping

- **Training**
  Learn relationship between linguistic & acoustic features

Linguistic → acoustic mapping

- **Training**
  Learn relationship between linguistic & acoustic features

- **Synthesis**
  Map linguistic features to acoustic ones

Linguistic → acoustic mapping

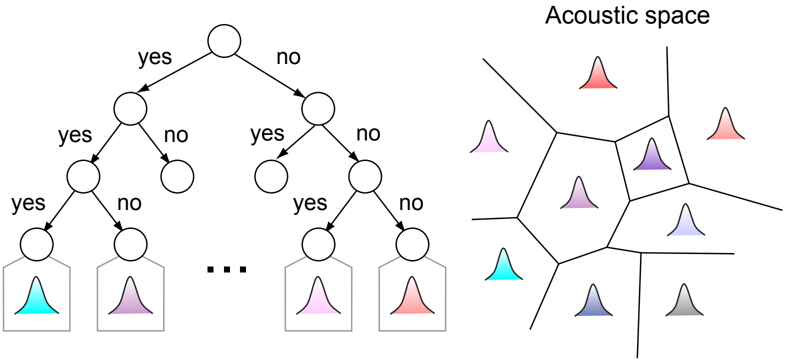
- **Training**
  Learn relationship between linguistic & acoustic features

- **Synthesis**
  Map linguistic features to acoustic ones

- **Linguistic features used in SPSS**
  - Phoneme, syllable, word, phrase, utterance-level features
  - e.g., phone identity, POS, stress, # of words in a phrase
  - Around 50 different types, much more than ASR (typically 3–5)

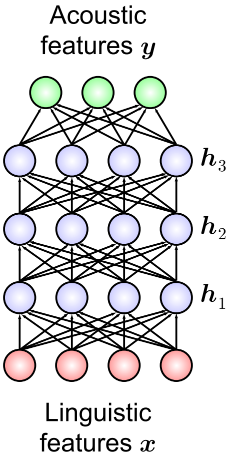
Effective modeling is essential

HMM-based acoustic modeling for SPSS [4]



Decision tree-clustered HMM w/ GMM state-output distributions

NN-based acoustic modeling for SPSS [20]



NN output $\rightarrow \mathbb{E}[\boldsymbol{y}_t \mid \boldsymbol{x}_t] \rightarrow$ replace decision trees & GMMs

Advantages of NN-based acoustic modeling for SPSS

- **Integrating feature extraction**
  - Efficiently model high-dimensional, highly correlated features
  - Layered architecture w/ non-linear operations
    $\longrightarrow$ Integrated linguistic feature extraction to acoustic modeling

Advantages of NN-based acoustic modeling for SPSS

- **Integrating feature extraction**
  - Efficiently model high-dimensional, highly correlated features
  - Layered architecture w/ non-linear operations
    $\longrightarrow$ Integrated linguistic feature extraction to acoustic modeling

- **Distributed representation**
  More efficient than localist one if data has componential structure
  $\longrightarrow$ Better modeling / Fewer parameters

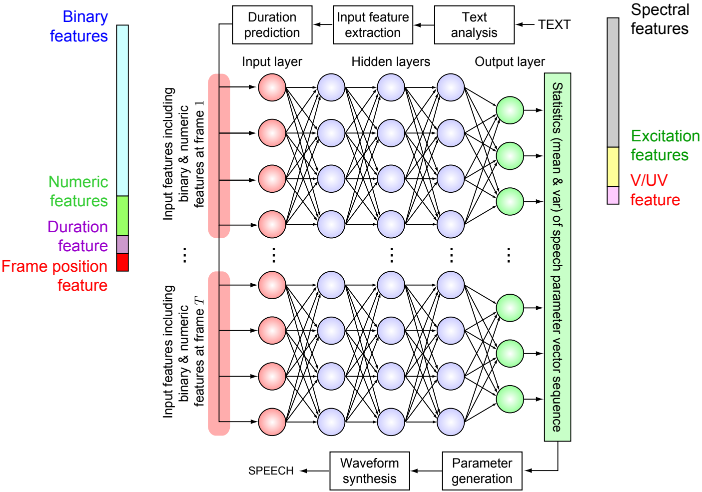
Advantages of NN-based acoustic modeling for SPSS

- **Integrating feature extraction**
  - Efficiently model high-dimensional, highly correlated features
  - Layered architecture w/ non-linear operations
    $\longrightarrow$ Integrated linguistic feature extraction to acoustic modeling

- **Distributed representation**
  More efficient than localist one if data has componential structure
  $\longrightarrow$ Better modeling / Fewer parameters

- **Layered hierarchical structure in speech production**
  concept $\rightarrow$ linguistic $\rightarrow$ articulatory $\rightarrow$ vocal tract $\rightarrow$ waveform

Framework



Binary features
Numeric features
Duration feature
Frame position feature
Duration prediction
Input feature extraction
Text analysis
TEXT
Input layer
Hidden layers
Output layer
Input features including binary & numeric features at frame 1
Input features including binary & numeric features at frame T
Statistics (mean & var) of speech parameter vector sequence
SPEECH
Waveform synthesis
Parameter generation
Spectral features
Excitation features
V/UV feature

Framework

Is this new? . . . no

- NN [21]
- RNN [22]

Framework

Is this new? . . . no

- NN [21]
- RNN [22]

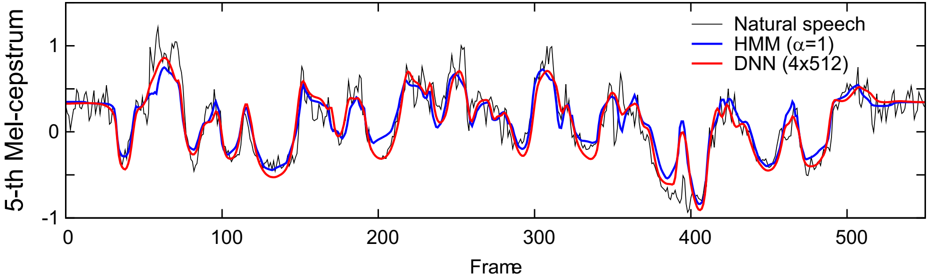
What's the difference?

- More layers, data, computational resources
- Better learning algorithm
- Statistical parametric speech synthesis techniques

Experimental setup

Database	US English female speaker
Training / test data	33000 & 173 sentences
Sampling rate	16 kHz
Analysis window	25-ms width / 5-ms shift
Linguistic features	11 categorical features 25 numeric features
Acoustic features	0–39 mel-cepstrum $\log F_0$, 5-band aperiodicity, Δ, Δ^2
HMM topology	5-state, left-to-right HSMM [23], MSD F_0 [24], MDL [25]
DNN architecture	1–5 layers, 256/512/1024/2048 units/layer sigmoid, continuous F_0 [26]
Preprocessing	Removed 80% of silence
Postprocessing	Postfiltering in cepstrum domain [15]

Example of speech parameter trajectories



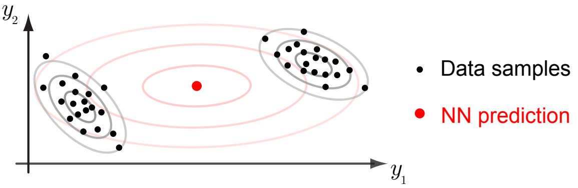
Subjective evaluations

Compared HMM- & DNN-based systems w/ similar # of params

- Paired comparison test
- 173 test sentences, 5 subjects per pair
- Up to 30 pairs per subject
- Crowd-sourced

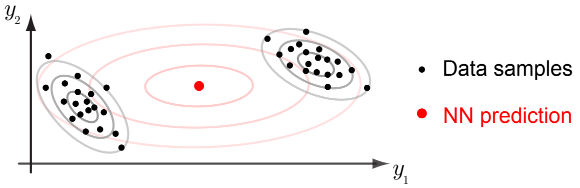
HMM (α)	DNN (#layers $\times$ #units)	Neutral	p value	z value
15.8 (16)	**38.5** (4 $\times$ 256)	45.7	$< 10^{-6}$	-9.9
16.1 (4)	**27.2** (4 $\times$ 512)	56.8	$< 10^{-6}$	-5.1
12.7 (1)	**36.6** (4 $\times$ 1 024)	50.7	$< 10^{-6}$	-11.5

Limitations of DNN-based acoustic modeling



- **Unimodality**
  - Human can speak in different ways → one-to-many mapping
  - NN trained by MSE loss → approximates conditional mean

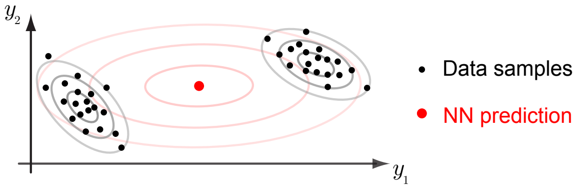
Limitations of DNN-based acoustic modeling



- **Unimodality**
  - Human can speak in different ways $\rightarrow$ one-to-many mapping
  - NN trained by MSE loss $\rightarrow$ approximates conditional mean

- **Lack of variance**
  - DNN-based SPSS uses variances computed from all training data
  - Parameter generation algorithm utilizes variances

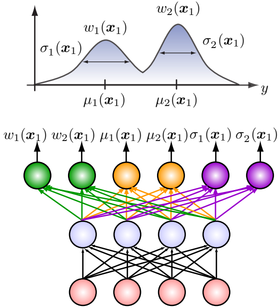
Limitations of DNN-based acoustic modeling



- **Unimodality**
  - Human can speak in different ways → one-to-many mapping
  - NN trained by MSE loss → approximates conditional mean
- **Lack of variance**
  - DNN-based SPSS uses variances computed from all training data
  - Parameter generation algorithm utilizes variances

Linear output layer → Mixture density output layer [27]

Mixture density network [27]



1-dim, 2-mix MDN

Inputs of activation function

$$z_j = \sum_{i=1}^{4} h_i w_{ij}$$

: Weights → Softmax activation function

$$w_1(\boldsymbol{x}) = \frac{\exp(z_1)}{\sum_{m=1}^{2} \exp(z_m)} \qquad w_2(\boldsymbol{x}) = \frac{\exp(z_2)}{\sum_{m=1}^{2} \exp(z_m)}$$

: Means → Linear activation function

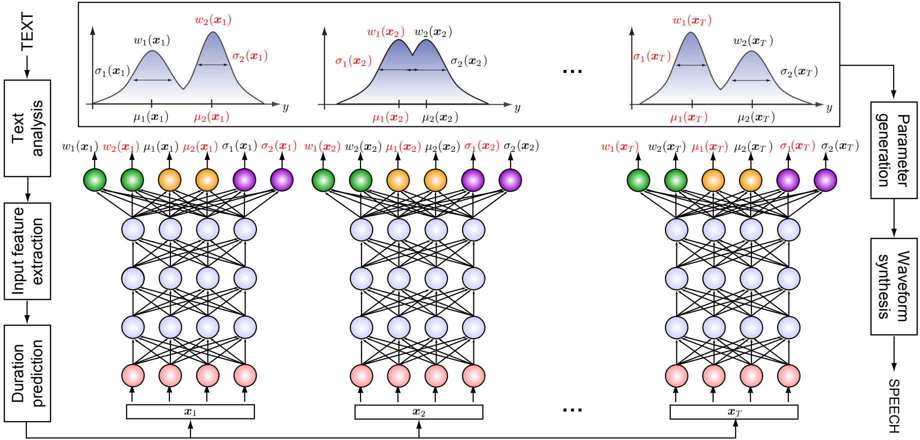
$$\mu_1(\boldsymbol{x}) = z_3 \qquad \mu_1(\boldsymbol{x}) = z_4$$

: Variances → Exponential activation function

$$\sigma_1(\boldsymbol{x}) = \exp(z_5) \qquad \sigma_2(\boldsymbol{x}) = \exp(z_6)$$

NN + mixture model (GMM)
→ NN outputs GMM weights, means, & variances

DMDN-based SPSS [28]



Experimental setup

- Almost the same as the previous setup
- Differences:

DNN architecture	4–7 hidden layers, 1024 units/hidden layer ReLU (hidden) / Linear (output)
DMDN architecture	4 hidden layers, 1024 units/ hidden layer ReLU [29] (hidden) / Mixture density (output) 1–16 mix
Optimization	AdaDec [30] (variant of AdaGrad [31]) on GPU

Subjective evaluation

- 5-scale mean opinion score (MOS) test (1: unnatural – 5: natural)
- 173 test sentences, 5 subjects per pair
- Up to 30 pairs per subject
- Crowd-sourced

HMM	1 mix	**3.54 ± 0.113**
	2 mix	3.40 ± 0.115
DNN	4×1024	3.64 ± 0.127
	5×1024	**3.68 ± 0.109**
	6×1024	3.65 ± 0.108
	7×1024	3.64 ± 0.129
DMDN (4×1024)	1 mix	3.65 ± 0.117
	2 mix	3.80 ± 0.107
	4 mix	3.77 ± 0.113
	8 mix	**3.81 ± 0.113**
	16 mix	3.79 ± 0.102

Limitations of DNN/MDN-based acoustic modeling

Fixed time span for input features

- Fixed number of preceding / succeeding contexts
- Difficult to incorporate long time span contextual effect

Limitations of DNN/MDN-based acoustic modeling

Fixed time span for input features

- Fixed number of preceding / succeeding contexts
- Difficult to incorporate long time span contextual effect

Frame-by-frame mapping

- Each frame is mapped independently
- Smoothing is still essential

Preference score (%)		
DNN w/ dyn	DNN w/o dyn	No pref
67.8	12.0	20.0

Limitations of DNN/MDN-based acoustic modeling

Fixed time span for input features

- Fixed number of preceding / succeeding contexts
- Difficult to incorporate long time span contextual effect

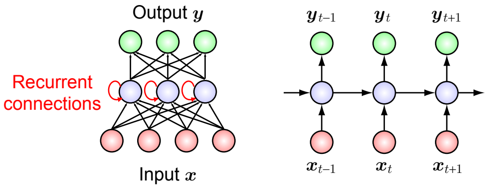
Frame-by-frame mapping

- Each frame is mapped independently
- Smoothing is still essential

Preference score (%)		
DNN w/ dyn	DNN w/o dyn	No pref
67.8	12.0	20.0

Recurrent connections → Recurrent NN (RNN) [32]

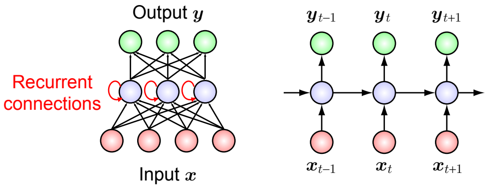
Simple Recurrent Network (SRN)



SRN-based acoustic modeling

$$\boldsymbol{h}_t = f\left(\boldsymbol{W}_{hx}\boldsymbol{x}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h\right), \qquad \boldsymbol{y}_t = \phi\left(\boldsymbol{W}_{yh}\boldsymbol{h}_t + \boldsymbol{b}_y\right)$$

Simple Recurrent Network (SRN)



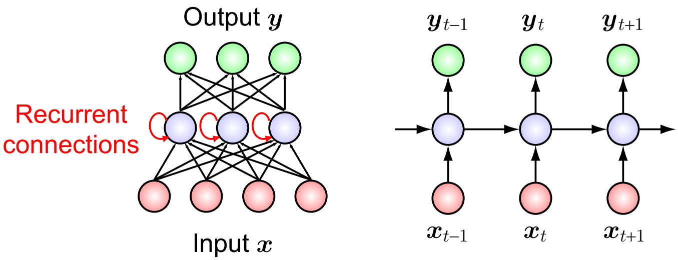
SRN-based acoustic modeling

$$\boldsymbol{h}_t = f\left(\boldsymbol{W}_{hx}\boldsymbol{x}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h\right), \qquad \boldsymbol{y}_t = \phi\left(\boldsymbol{W}_{yh}\boldsymbol{h}_t + \boldsymbol{b}_y\right)$$

With squared loss...

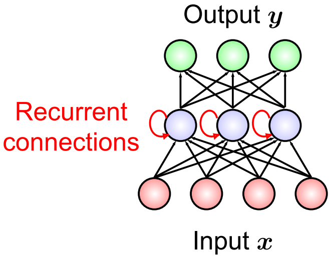
- DNN output (prediction) $\hat{\boldsymbol{y}}_t \rightarrow \mathbb{E}\left[\boldsymbol{y}_t \mid \boldsymbol{x}_t\right]$
- RNN output (prediction) $\hat{\boldsymbol{y}}_t \rightarrow \mathbb{E}\left[\boldsymbol{y}_t \mid \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\right]$

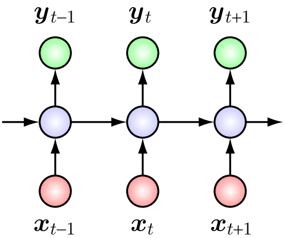
Simple Recurrent Network (SRN)



- **Only able to use previous contexts**
  $\rightarrow$ bidirectional RNN [32]

Simple Recurrent Network (SRN)





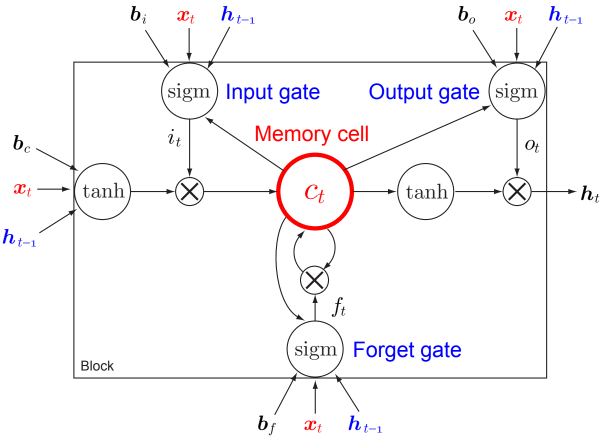
- **Only able to use previous contexts**
  $\longrightarrow$ bidirectional RNN [32]

- **Trouble accessing long-range contexts**
  - Information in hidden layers loops through recurrent connections
    $\rightarrow$ Quickly decay over time
  - Prone to being overwritten by new information arriving from inputs

  $\longrightarrow$ long short-term memory (LSTM) RNN [33]

Long short-term memory (LSTM) [33]

- RNN architecture designed to have better memory
- Uses linear **memory cells** surrounded by multiplicative gate units



Gate output: 0 -- 1

Input gate == 1
→ Write memory

Forget gate == 0
→ Reset memory

Output gate == 1
→ Read memory

Advantages of RNN-based acoustic modeling for SPSS

- **Model dependency between frames**
  - HMM: discontinuous (step-wise) $\longrightarrow$ *smoothing*
  - DNN: discontinuous (frame-by-frame mapping) [34] $\longrightarrow$ *smoothing*
  - RNN: smooth [35, 34]

Advantages of RNN-based acoustic modeling for SPSS

- **Model dependency between frames**
  - HMM: discontinuous (step-wise) $\longrightarrow$ *smoothing*
  - DNN: discontinuous (frame-by-frame mapping) [34] $\longrightarrow$ *smoothing*
  - RNN: smooth [35, 34]

- **Low latency**
  - Unidirectional structure allows fully frame-level streaming [34]

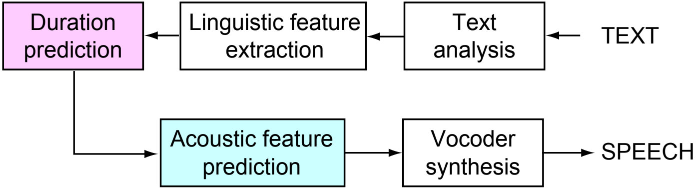
Advantages of RNN-based acoustic modeling for SPSS

- **Model dependency between frames**
  - HMM: discontinuous (step-wise) $\longrightarrow$ *smoothing*
  - DNN: discontinuous (frame-by-frame mapping) [34] $\longrightarrow$ *smoothing*
  - RNN: smooth [35, 34]

- **Low latency**
  - Unidirectional structure allows fully frame-level streaming [34]

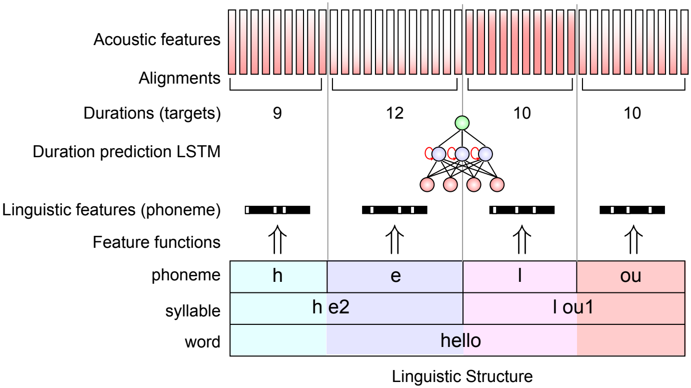
- **More efficient representation**
  - RNN offers more efficient representation than DNN for time series

Synthesis pipeline



Duration & acoustic feature prediction blocks involve NN

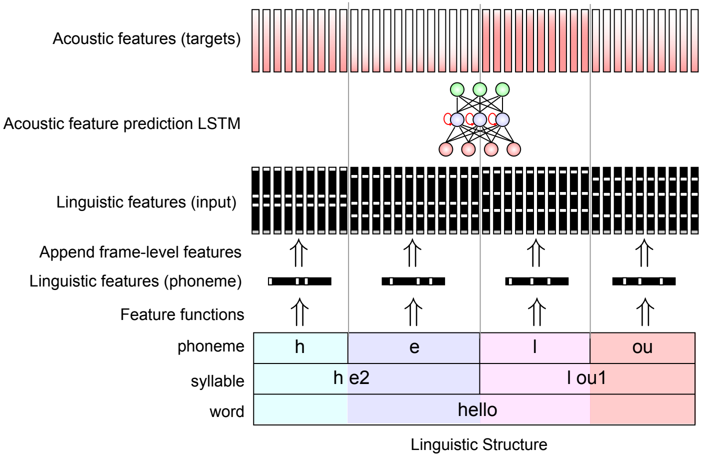
Duration modeling



Feature function examples

phoneme == 'h' ?   syllable stress == '2' ?   # of syllables in word?

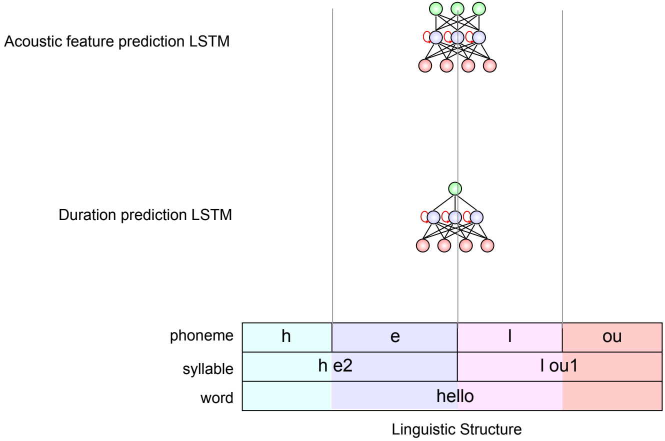
Acoustic modeling



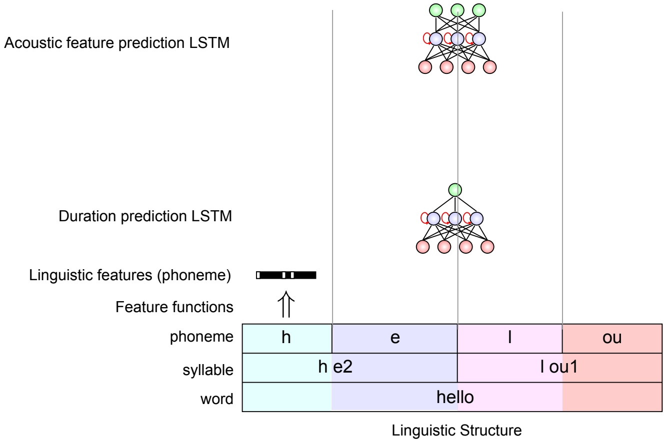
Append frame-level features

Relative position of frame in phoneme

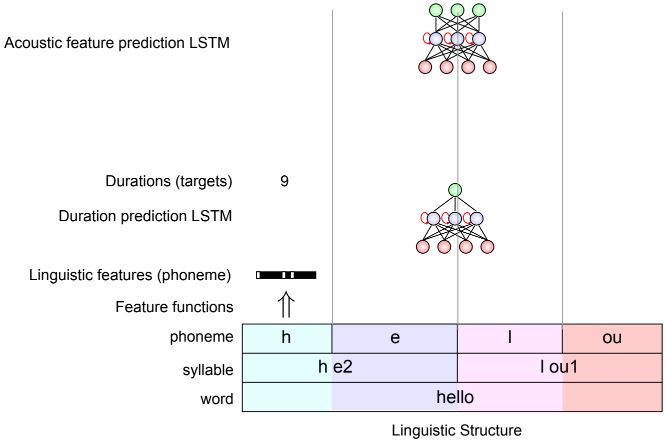
Streaming synthesis



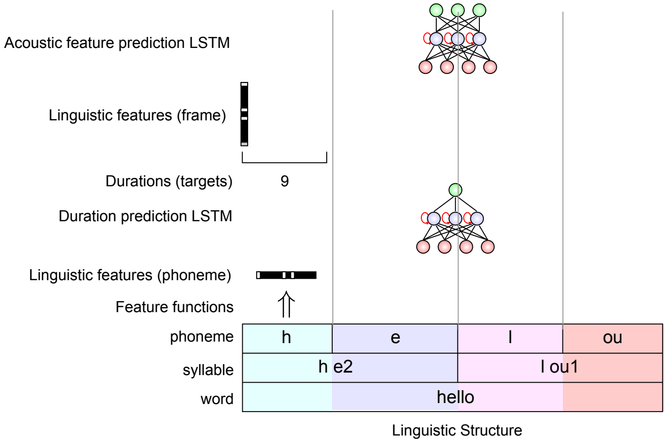
Streaming synthesis



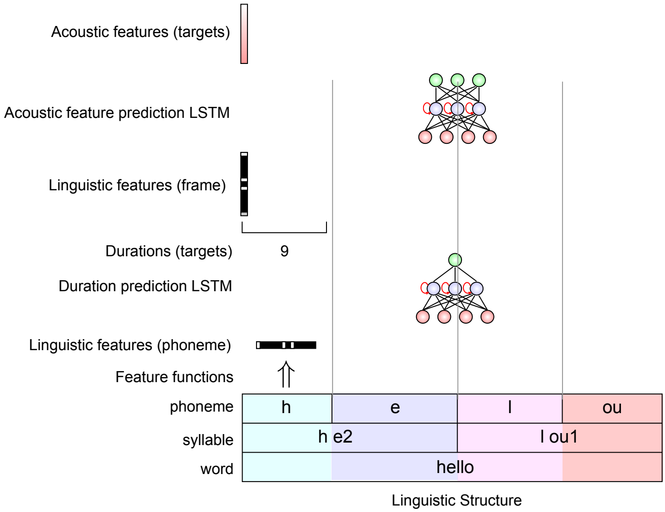
Streaming synthesis



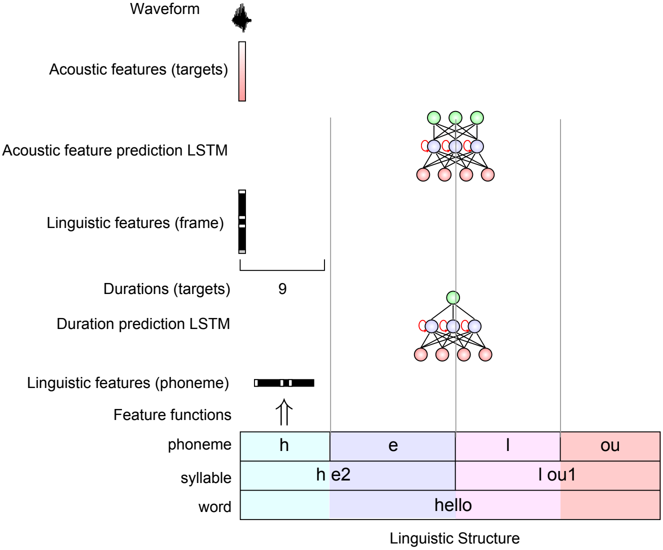
Streaming synthesis



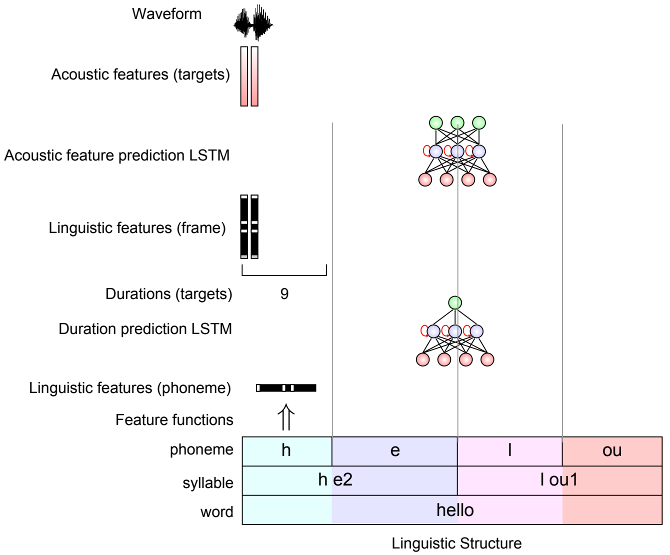
Streaming synthesis



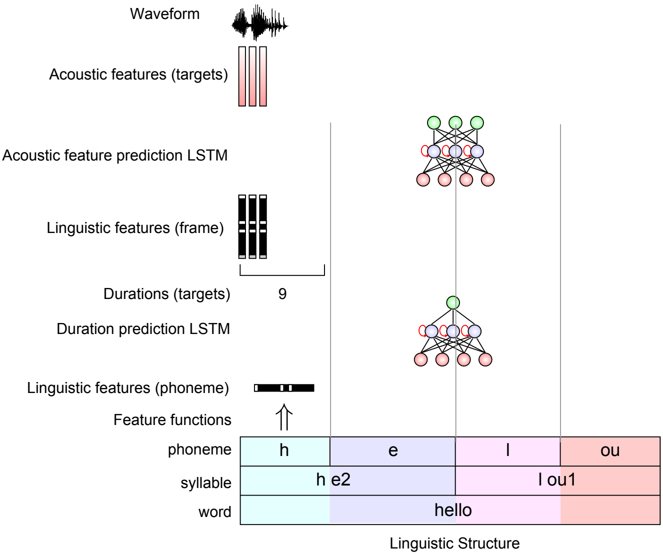
Streaming synthesis



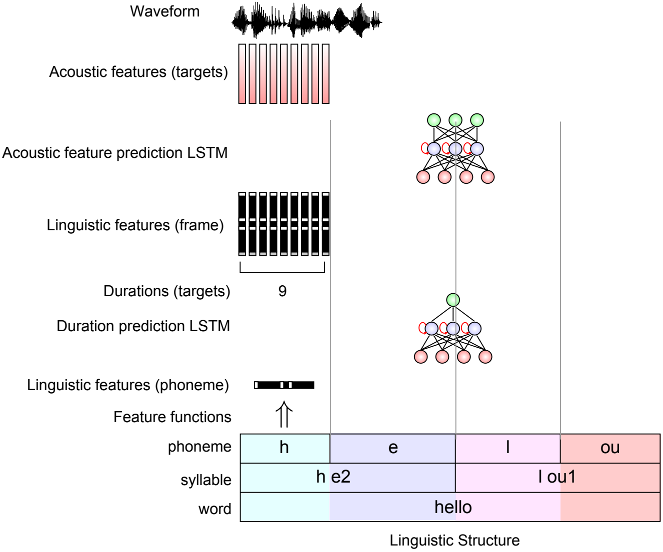
Streaming synthesis



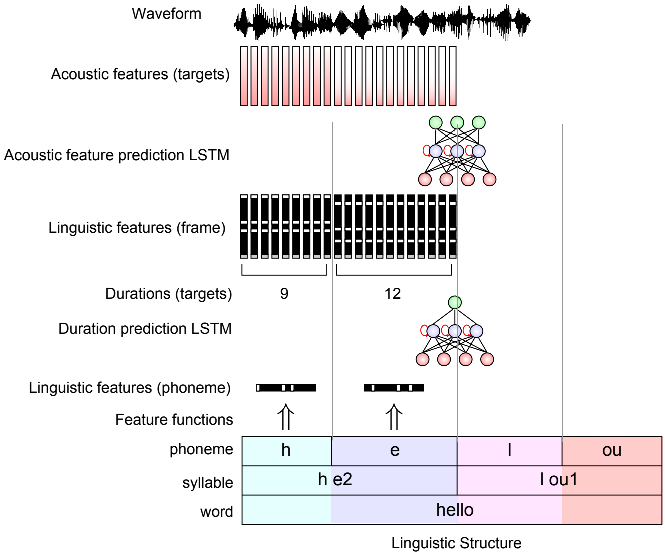
Streaming synthesis



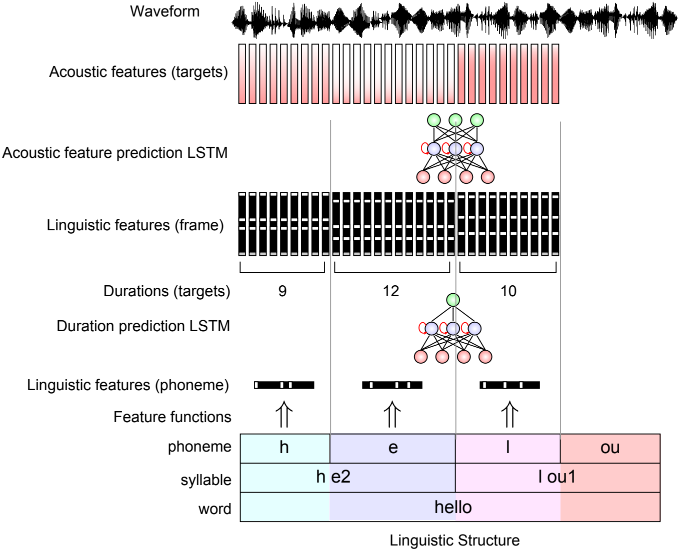
Streaming synthesis



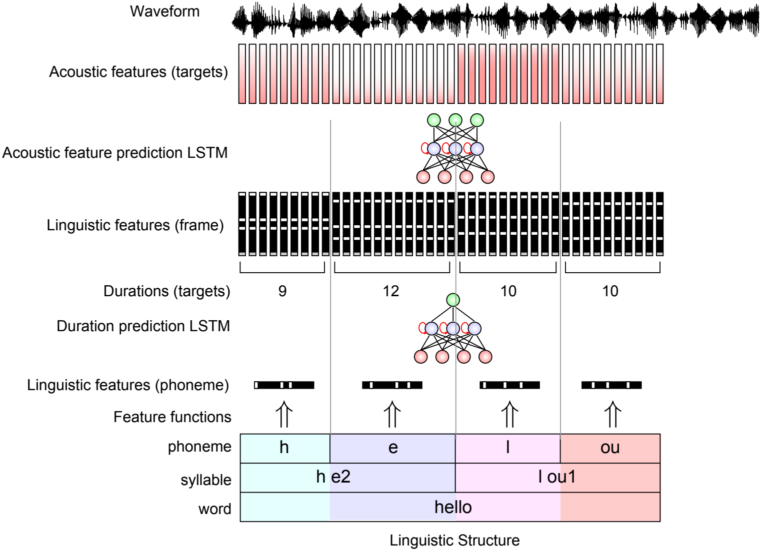
Streaming synthesis



Streaming synthesis



Streaming synthesis



Data & speech analysis

Database	US English female speaker 34 632 utterances
Speech analysis	16 kHz sampling 25-ms width / 5-ms shift
Synthesis	Vocaine [36] Postfiltering-based enhancement
Input	DNN: 442 linguistic features ULSTM: 291 linguistic features
Target	0–39 mel-cepstrum features continuous $\log F_0$ [26] 5-band aperiodicity optionally Δ, Δ^2

Training

Preprocessing	Acoustic: removed 80% silence Duration: removed first/last silence
Normalization	Input: mean / standard deviations Output: 0.01 – 0.99
Architecture	DNN: 4 $\times$ 1024 units, ReLU [29] ULSTM: 1 $\times$ 256 cells
Output layer	Acoustic: feed-forward or recurrent Duration: feed-forward
Initialization	DNN: random + layer-wise BP [37] ULSTM: random
Optimization	Common: squared loss, SGD DNN: GPU, AdaDec [38] ULSTM: distributed CPU [39]

Subjective tests

Common	100 sentences Crowd-sourcing Using head-phones
MOS	7 evaluations per sample Up to 30 stimuli per subject 5-scale score in naturalness (1: Bad – 5: Excellent)
Preference	5 evaluations per pair Up to 30 pairs per subject Chose prefered one or "neutral"

of future contexts

# of future contexts	5-scale MOS
0	3.57 ± 0.121
1	3.75 ± 0.119
2	$\mathbf{3.81} \pm 0.115$
3	3.78 ± 0.118
4	3.75 ± 0.115

Preference scores

DNN		ULSTM				
Feed-forward		Feed-forward		Recurrent		Neutral
w/	w/o	w/	w/o	w/	w/o	
67.8	12.0					20.0
18.4		**34.9**				47.6
		21.0	12.2			66.8
		21.8			21.0	57.2
				16.6	**29.2**	54.2

MOS

- DNN w/ dynamic features
- ULSTM w/o dynamic features, w/ recurrent output layer

Model	# params	5-scale MOS
DNN	3,747,979	3.37 ± 0.114
ULSTM	**476,435**	**3.72** $\pm$ 0.105

Latency

- Nexus 7 2013
- Use Advanced SIMD (NEON), single thread
- Audio buffer size: 1024
- HMM one used time-recursive version w/ $L = 15$
- HMM & ULSTM used the same text analysis front-end

	Average latency (ms)	
	HMM	ULSTM
chars	26	25
short	123	**55**
long	311	**115**

LSTM-based TTS demo

- Turkish
- Korean
- Mandarin
- Thai
- French
- Italian
- German
- Spanish
- Russian
- Polish
- Dutch
- Japanese
- US, UK, & Indian English

Outline

Basics of HMM-based speech synthesis
- Background
- HMM-based speech synthesis

Advanced topics in HMM-based speech synthesis
- Flexibility
- Improve naturalness

Neural network-based speech synthesis
- Feed-forward neural network (DNN & DMDN)
- Recurrent neural network (RNN & LSTM-RNN)
- Results

Conclusion

Summary

Statistical parametric speech synthesis

- **Vocoding + acoustic model**

- **HMM-based SPSS**
  - Flexible (e.g., adaptation, interpolation)
  - Improvements
    - Vocoding
    - Acoustic modeling
    - Oversmoothing compensation

- **NN-based SPSS**
  - Learn mapping from linguistic features to acoustic ones
  - Static network (DNN, DMDN) → dynamic ones (LSTM)

Google academic program

- **Award programs**
  - **Google Faculty Research Awards**
    Provides unrestricted gifts to support fulltime faculty members
  - **Google Focused Research Awards**
    Fund specific key research areas
  - **Visiting Faculty Program**
    Support full-time faculty in research areas of mutual interest

  http://research.google.com/university/relations/

- **Student support programs**
  - **Graduate Fellowships**
    Recognize outstanding graduate students
  - **Internships**
    Work on real-world problems with Google's data & infrastructure

  http://research.google.com/university/student-support/
  http://www.google.com/about/careers/students/

References I

[1] E. Moulines and F. Charpentier.
Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones.
Speech Commun., 9:453–467, 1990.

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

[4] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura.
Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis.
In *Proc. Eurospeech*, pages 2347–2350, 1999.

[5] F. Itakura and S. Saito.
A statistical method for estimation of speech spectral density and formant frequencies.
Trans. IEICE, J53–A:35–42, 1970.

[6] S. Imai.
Cepstral analysis synthesis on the mel frequency scale.
In *Proc. ICASSP*, pages 93–96, 1983.

[7] J. Odell.
The use of context in large vocabulary speech recognition.
PhD thesis, Cambridge University, 1995.

[8] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura.
Duration modeling for HMM-based speech synthesis.
In *Proc. ICSLP*, pages 29–32, 1998.

References II

[9] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura.
Speech parameter generation algorithms for HMM-based speech synthesis.
In *Proc. ICASSP*, pages 1315–1318, 2000.

[10] J. Yamagishi.
Average-Voice-Based Speech Synthesis.
PhD thesis, Tokyo Institute of Technology, 2006.

[11] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura.
Speaker interpolation in HMM-based speech synthesis system.
In *Proc. Eurospeech*, pages 2523–2526, 1997.

[12] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura.
Eigenvoices for HMM-based speech synthesis.
In *Proc. ICSLP*, pages 1269–1272, 2002.

[13] H. Zen, N. Braunschweiler, S. Buchholz, M. Gales, K. Knill, S. Krstulovic, and J. Latorre.
Statistical parametric speech synthesis based on speaker and language factorization.
IEEE Trans. Acoust. Speech Lang. Process., 20(6):1713–1724, 2012.

[14] T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi.
A style control technique for HMM-based expressive speech synthesis.
IEICE Trans. Inf. Syst., E90-D(9):1406–1413, 2007.

[15] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura.
Incorporation of mixed excitation model and postfilter into HMM-based text-to-speech synthesis.
IEICE Trans. Inf. Syst., J87-D-II(8):1563–1571, 2004.

[16] H. Kawahara, I. Masuda-Katsuse, and A.de Cheveigné.
Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f_0 extraction: possible role of a repetitive structure in sounds.
Speech Commun., 27:187–207, 1999.

References III

[17] R. Maia, T. Toda, H. Zen, Y. Nankaku, and K. Tokuda.
An excitation model for HMM-based speech synthesis based on residual modeling.
In *Proc. ISCA SSW6*, pages 131–136, 2007.

[18] H. Zen, K. Tokuda, and T. Kitamura.
Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic features.
Comput. Speech Lang., 21(1):153–173, 2007.

[19] T. Toda and K. Tokuda.
A speech parameter generation algorithm considering global variance for HMM-based speech synthesis.
IEICE Trans. Inf. Syst., E90-D(5):816–824, 2007.

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

[21] O. Karaali, G. Corrigan, and I. Gerson.
Speech synthesis with neural networks.
In *Proc. World Congress on Neural Networks*, pages 45–50, 1996.

[22] C. Tuerk and T. Robinson.
Speech synthesis using artificial network trained on cepstral coefficients.
In *Proc. Eurospeech*, pages 1713–1716, 1993.

[23] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura.
A hidden semi-Markov model-based speech synthesis system.
IEICE Trans. Inf. Syst., E90-D(5):825–834, 2007.

[24] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi.
Multi-space probability distribution HMM.
IEICE Trans. Inf. Syst., E85-D(3):455–464, 2002.

References IV

[25] K. Shinoda and T. Watanabe.
Acoustic modeling based on the MDL criterion for speech recognition.
In *Proc. Eurospeech*, pages 99–102, 1997.

[26] K. Yu and S. Young.
Continuous F0 modelling for HMM based statistical parametric speech synthesis.
IEEE Trans. Audio Speech Lang. Process., 19(5):1071–1079, 2011.

[27] C. Bishop.
Mixture density networks.
Technical Report NCRG/94/004, Neural Computing Research Group, Aston University, 1994.

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

[29] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q.-V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton.
On rectified linear units for speech processing.
In *Proc. ICASSP*, pages 3517–3521, 2013.

[30] A. Senior, G. Heigold, M. Ranzato, and K. Yang.
An empirical study of learning rates in deep neural networks for speech recognition.
In *Proc. ICASSP*, pages 6724–6728, 2013.

[31] J. Duchi, E. Hazan, and Y. Singer.
Adaptive subgradient methods for online learning and stochastic optimization.
The Journal of Machine Learning Research, pages 2121–2159, 2011.

[32] M. Schuster and K. Paliwal.
Bidirectional recurrent neural networks.
IEEE Trans. Signal Process., 45(11):2673–2681, 1997.

References V

[33] S. Hochreiter and J. Schmidhuber.
Long short-term memory.
Neural computation, 9(8):1735–1780, 1997.

[34] H. Zen and H. Sak.
Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis.
In *Proc. ICASSP*, pages 4470–4474, 2015.

[35] Y. Fan, Y. Qian, F. Xie, and F. Soong.
TTS synthesis with bidirectional LSTM based recurrent neural networks.
In *Proc. Interspeech*, 2014.
(Submitted) `http://research.microsoft.com/en-us/projects/dnntts/`.

[36] Y. Agiomyrgiannakis.
Vocaine the vocoder and applications is speech synthesis.
In *Proc. ICASSP*, pages 4230–4234, 2015.

[37] F. Seide, G. Li, X. Chen, and D. Yu.
Feature engineering in context-dependent deep neural networks for conversational speech transcription.
In *Proc. ASRU*, pages 24–29. IEEE, 2011.

[38] A. Senior, G. Heigold, M. Ranzato, and K. Yang.
An empirical study of learning rates in deep neural networks for speech recognition.
In *Proc. ICASSP*, pages 6724–6728, 2013.

[39] H. Sak, A. Senior, and F. Beaufays.
Long short-term memory recurrent neural network architectures for large scale acoustic modeling.
In *Proc. Interspeech*, 2014.