Statistical Parametric Speech Synthesis

Heiga Zen
Google

June 9th, 2014

Outline

Background
- HMM-based statistical parametric speech synthesis (SPSS)
- Flexibility
- Improvements

Statistical parametric speech synthesis with neural networks
- Deep neural network (DNN)-based SPSS
- Deep mixture density network (DMDN)-based SPSS
- Recurrent neural network (RNN)-based SPSS

Summary
- Summary

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

- **Automatic speech recognition (ASR)**
  Speech (continuous time series) → Text (discrete symbol sequence)

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

- **Automatic speech recognition (ASR)**
  Speech (continuous time series) → Text (discrete symbol sequence)

- **Machine translation (MT)**
  Text (discrete symbol sequence) → Text (discrete symbol sequence)

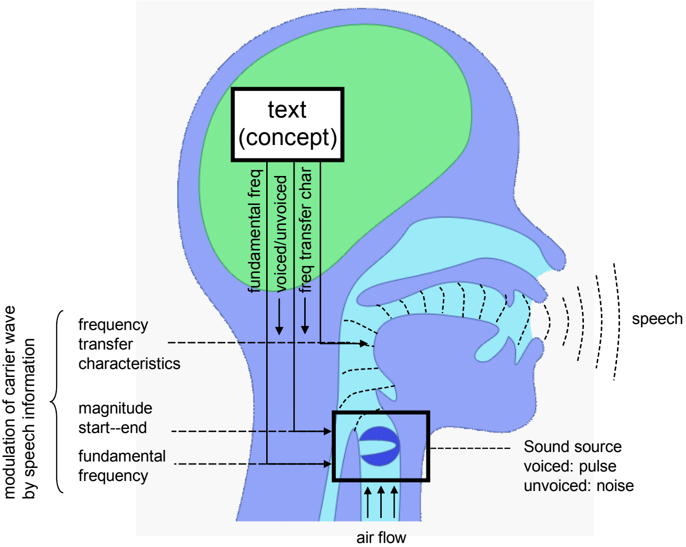
Text-to-speech as sequence-to-sequence mapping

- **Automatic speech recognition (ASR)**
  Speech (continuous time series) → Text (discrete symbol sequence)

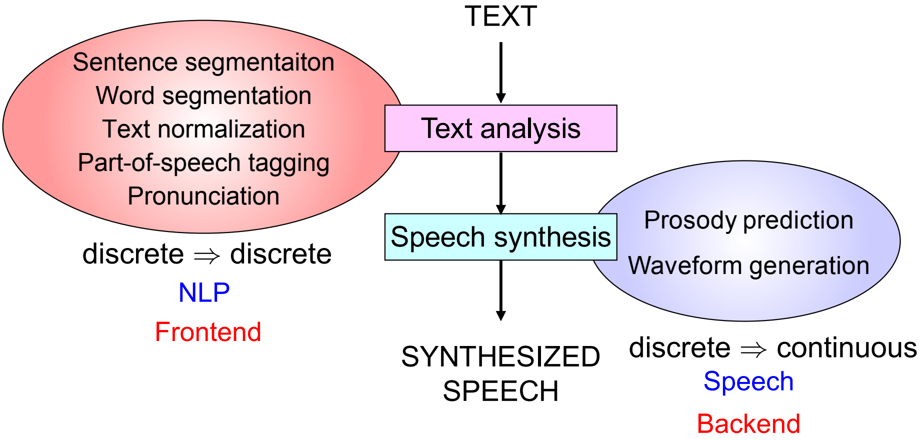
- **Machine translation (MT)**
  Text (discrete symbol sequence) → Text (discrete symbol sequence)

- **Text-to-speech synthesis (TTS)**
  Text (discrete symbol sequence) → Speech (continuous time series)

Speech production process



Typical flow of TTS system



TEXT

Text analysis

Sentence segmentaiton
Word segmentation
Text normalization
Part-of-speech tagging
Pronunciation

discrete ⇒ discrete

NLP

Frontend

Speech synthesis

Prosody prediction
Waveform generation

discrete ⇒ continuous

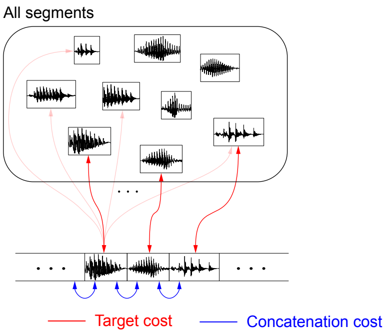
Speech

Backend

SYNTHESIZED SPEECH

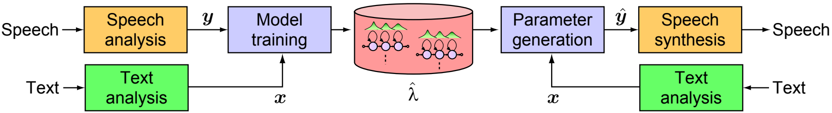
This talk focuses on backend

Concatenative 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]

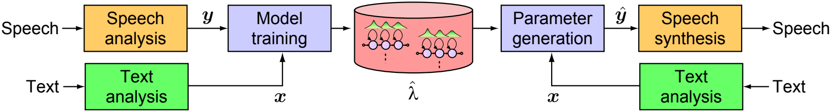
Statistical parametric speech synthesis (SPSS) [3]



- **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 $(\boldsymbol{x}, \boldsymbol{y})$

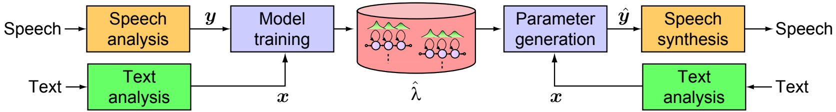
$$\hat{\lambda} = \arg\max p(\boldsymbol{y} \mid \boldsymbol{x}, \lambda)$$

- **Synthesis**
  - Extract $\boldsymbol{x}$ from text to be synthesized
  - Generate most probable $\boldsymbol{y}$ from $\hat{\lambda}$

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

  - Reconstruct speech from $\hat{\boldsymbol{y}}$

Statistical parametric speech synthesis (SPSS) [3]



- Large data + automatic training
  → Automatic voice building
- Parametric representation of speech
  → Flexible to change its voice characteristics

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

Outline

Background

- HMM-based statistical parametric speech synthesis (SPSS)
- Flexibility
- Improvements

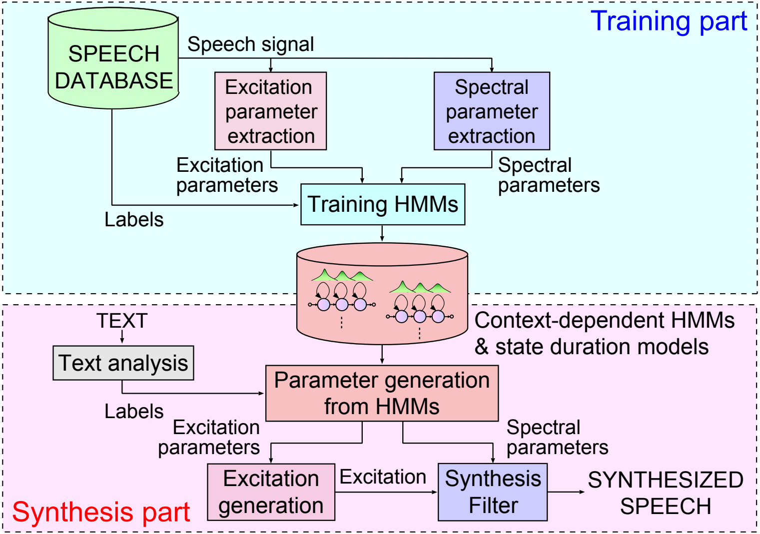
Statistical parametric speech synthesis with neural networks

- Deep neural network (DNN)-based SPSS
- Deep mixture density network (DMDN)-based SPSS
- Recurrent neural network (RNN)-based SPSS

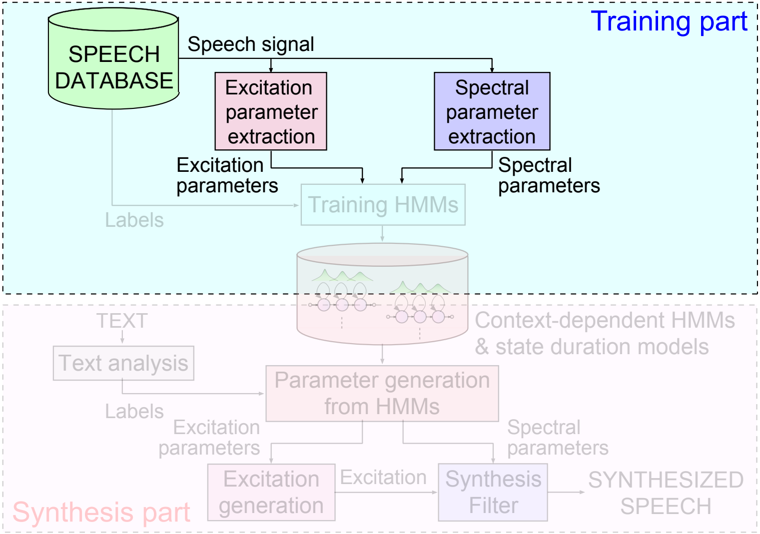
Summary

- Summary

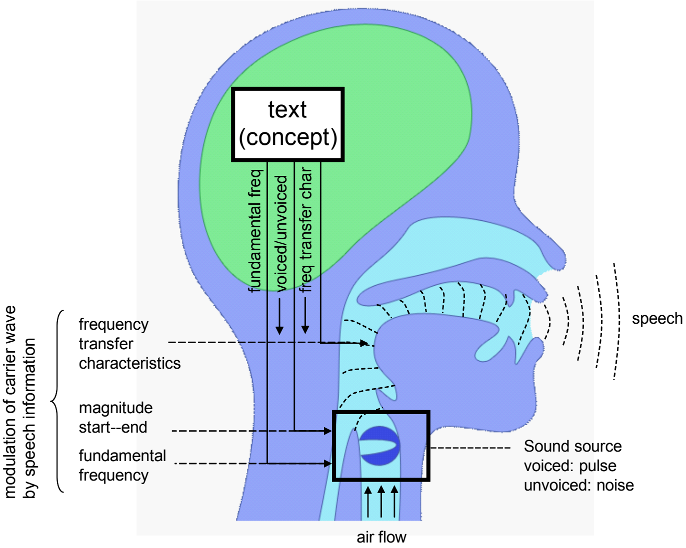
HMM-based speech synthesis [4]



HMM-based speech synthesis [4]



Speech production process



text (concept)

fundamental freq

voiced/unvoiced

freq transfer char

modulation of carrier wave by speech information

frequency transfer characteristics

magnitude start--end

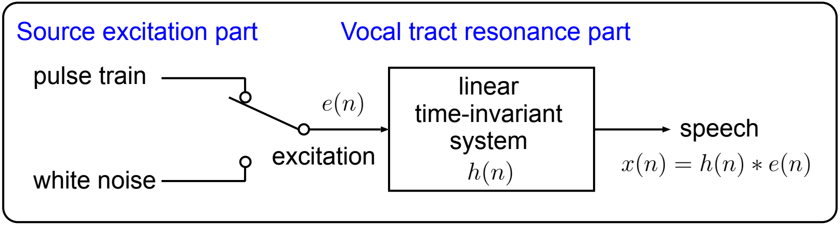
fundamental frequency

speech

Sound source
voiced: pulse
unvoiced: noise

air flow

Source-filter model



Source excitation part

Vocal tract resonance part

pulse train

white noise

$e(n)$

excitation

linear
time-invariant
system
$h(n)$

speech

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

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

$\downarrow$ 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

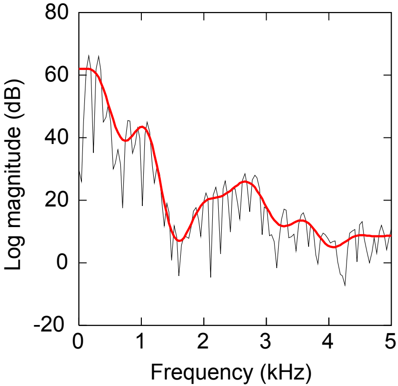
Autoregressive (AR) model	**Exponential (EX) model**
$H(z) = \dfrac{K}{1 - \displaystyle\sum_{m=0}^{M} c(m) z^{-m}}$	$H(z) = \exp \displaystyle\sum_{m=0}^{M} c(m) z^{-m}$

Estimate model parameters based on ML

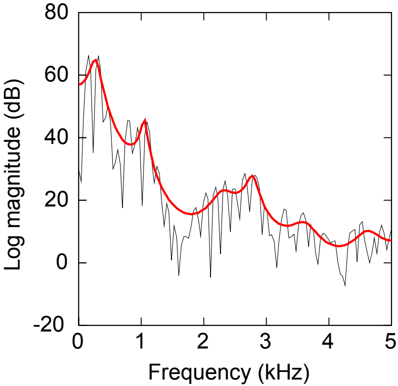
$$\boldsymbol{c} = \arg\max_{\boldsymbol{c}} p(\boldsymbol{x} \mid \boldsymbol{c})$$

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

Examples of speech spectra

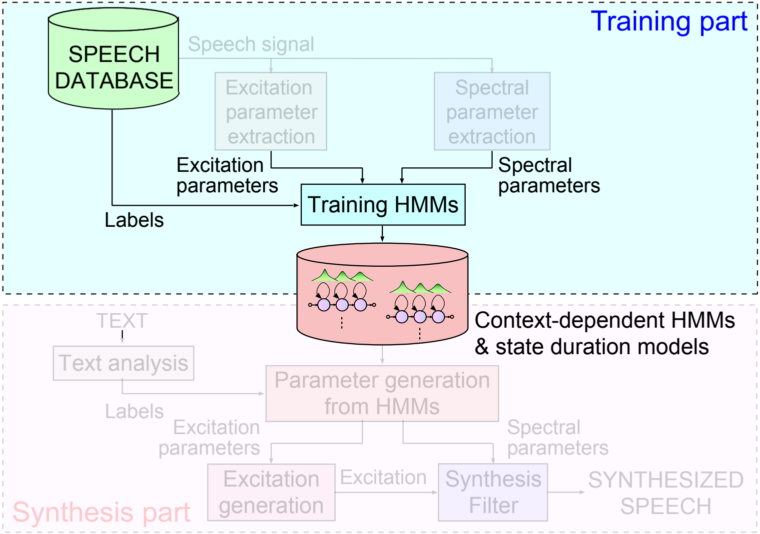


(a) ML-based cepstral analysis



(b) Linear prediction

HMM-based speech synthesis [4]



Training part

SPEECH DATABASE

Speech signal

Excitation parameter extraction

Spectral parameter extraction

Excitation parameters

Spectral parameters

Labels

Training HMMs

Context-dependent HMMs & state duration models

TEXT

Text analysis

Labels

Parameter generation from HMMs

Excitation parameters

Spectral parameters

Excitation generation

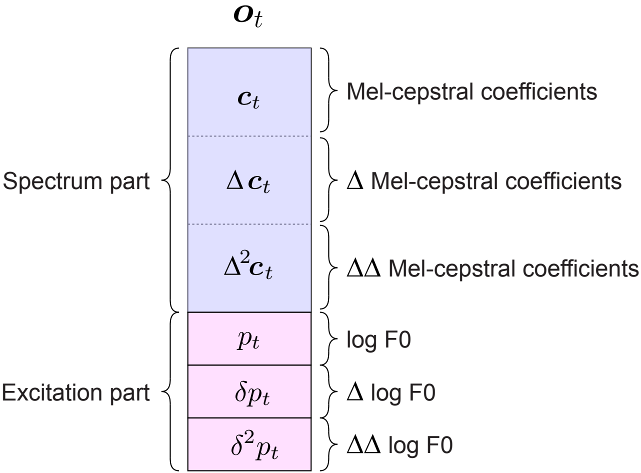
Excitation

Synthesis Filter

SYNTHESIZED SPEECH

Synthesis part

Structure of state-output (observation) vectors



$\boldsymbol{o}_t$

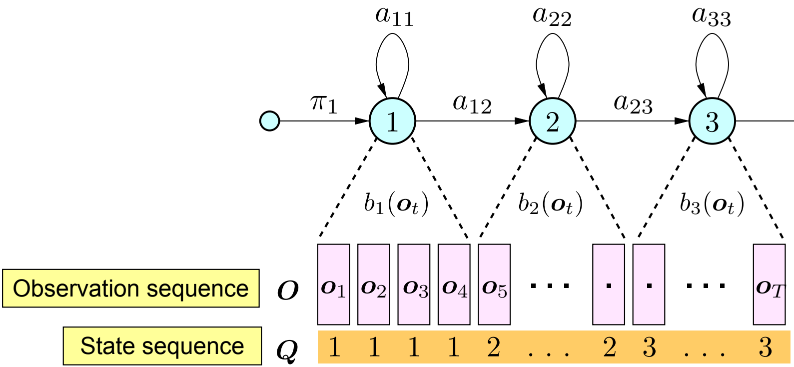
Spectrum part

- $\boldsymbol{c}_t$: Mel-cepstral coefficients
- $\Delta\,\boldsymbol{c}_t$: Δ Mel-cepstral coefficients
- $\Delta^2\boldsymbol{c}_t$: ΔΔ Mel-cepstral coefficients

Excitation part

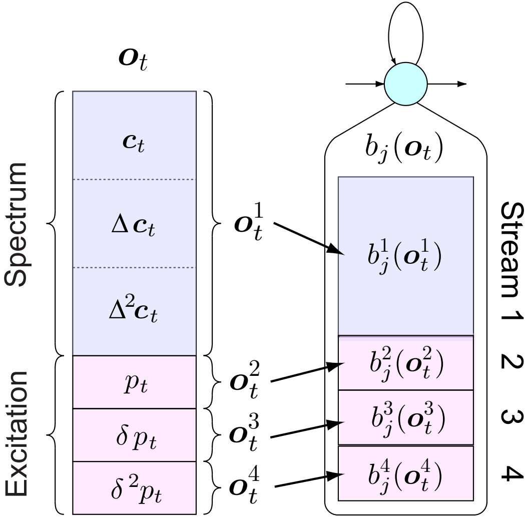
- p_t : log F0
- δp_t : Δ log F0
- $\delta^2 p_t$: ΔΔ log F0

Hidden Markov model (HMM)

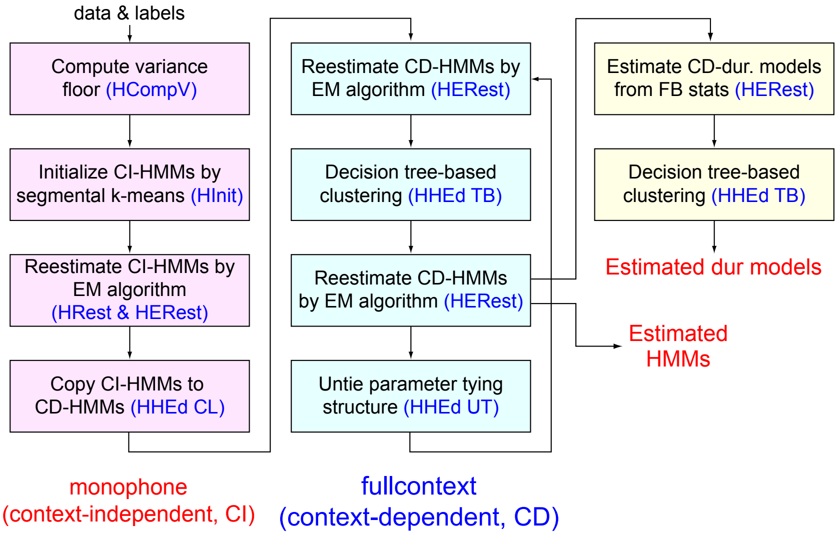


Multi-stream HMM structure

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



Training process



data & labels

Compute variance floor (HCompV)

Initialize CI-HMMs by segmental k-means (HInit)

Reestimate CI-HMMs by EM algorithm (HRest & HERest)

Copy CI-HMMs to CD-HMMs (HHEd CL)

Reestimate CD-HMMs by EM algorithm (HERest)

Decision tree-based clustering (HHEd TB)

Reestimate CD-HMMs by EM algorithm (HERest)

Untie parameter tying structure (HHEd UT)

Estimate CD-dur. models from FB stats (HERest)

Decision tree-based clustering (HHEd TB)

Estimated dur models

Estimated HMMs

monophone (context-independent, CI)

fullcontext (context-dependent, CD)

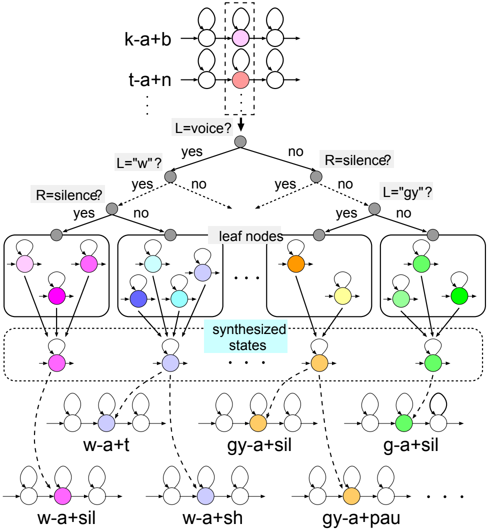
Context-dependent acoustic modeling

- {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

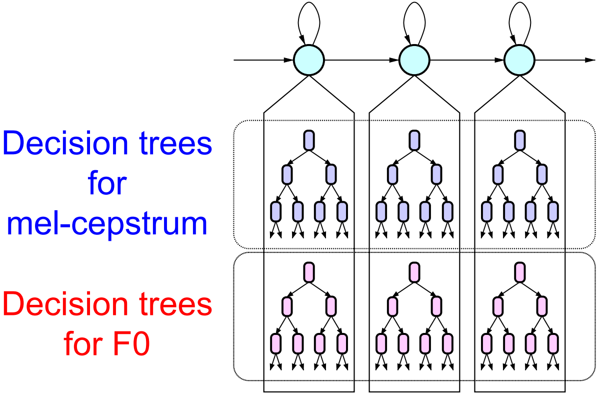
. . .

Impossible to have all possible models

Decision tree-based state clustering [7]

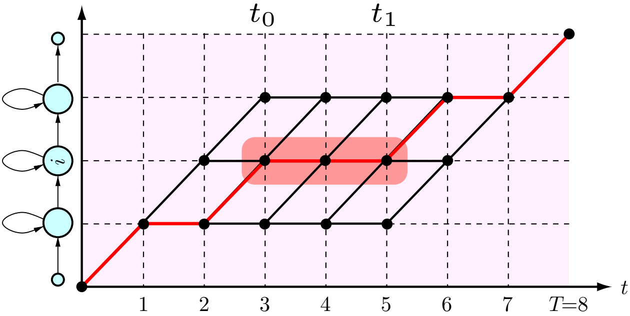


Stream-dependent tree-based clustering



Spectrum & excitation can have different context dependency
→ Build decision trees individually

State duration models [8]

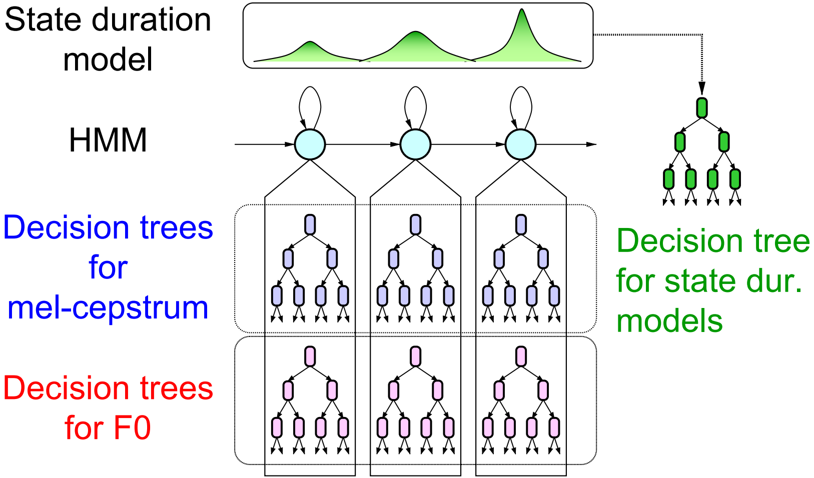


Probability to enter state i at t_0 then leave at $t_1 + 1$

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

→ estimate state duration models

Stream-dependent tree-based clustering



State duration model

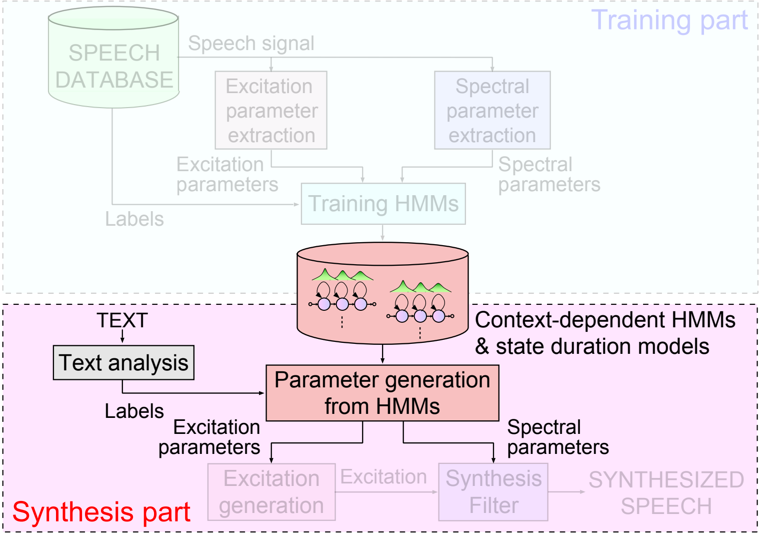
HMM

Decision trees for mel-cepstrum

Decision trees for F0

Decision tree for state dur. models

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]

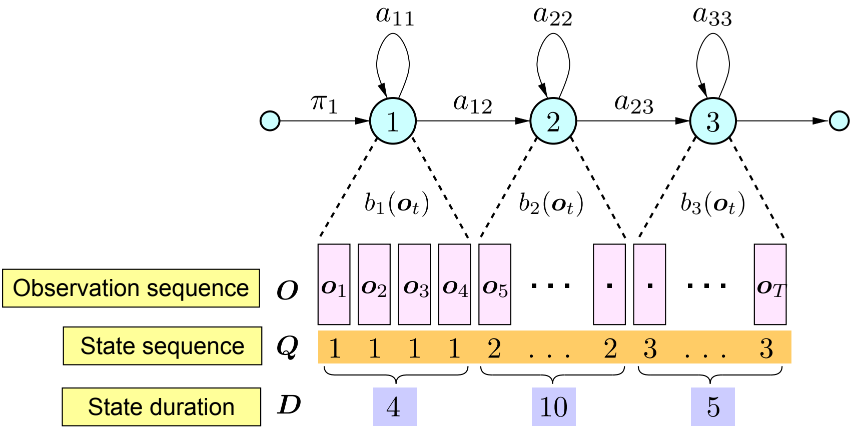
Generate most probable state outputs given HMM and words

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

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



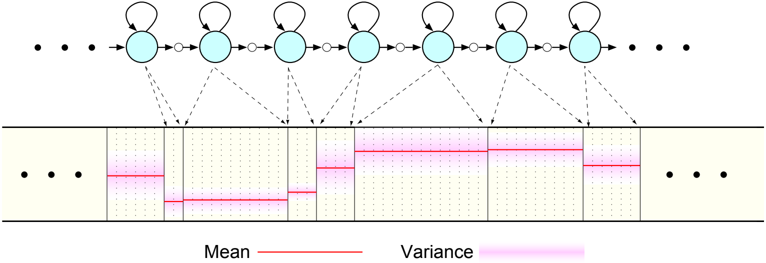
Observation sequence

State sequence

State duration

Best state outputs

w/o dynamic features

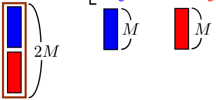


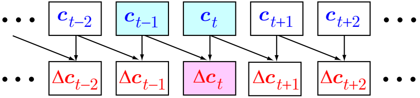
$\hat{o}$ becomes step-wise mean vector sequence

Using dynamic features

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

$$\begin{array}{cc} & \boldsymbol{o} \\ & \left[\begin{array}{c} \vdots \\ \hline \boldsymbol{c}_{t-1} \\ \Delta\boldsymbol{c}_{t-1} \\ \hline \boldsymbol{c}_{t} \\ \Delta\boldsymbol{c}_{t} \\ \hline \boldsymbol{c}_{t+1} \\ \Delta\boldsymbol{c}_{t+1} \\ \hline \vdots \end{array}\right] \end{array} \begin{array}{c} \\ \\ \boldsymbol{o}_{t-1} \\ \\ \boldsymbol{o}_{t} \\ \\ \boldsymbol{o}_{t+1} \\ \\ \end{array} = \begin{array}{c} \boldsymbol{W} \\ \left[\begin{array}{cccccc} \cdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ \hline \cdots & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \cdots \\ \cdots & -\boldsymbol{I} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \cdots \\ \hline \cdots & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \cdots \\ \cdots & \mathbf{0} & -\boldsymbol{I} & \boldsymbol{I} & \mathbf{0} & \cdots \\ \hline \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \cdots \\ \cdots & \mathbf{0} & \mathbf{0} & -\boldsymbol{I} & \boldsymbol{I} & \cdots \\ \hline \cdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{array}\right] \end{array} \begin{array}{c} \boldsymbol{c} \\ \left[\begin{array}{c} \vdots \\ \hline \boldsymbol{c}_{t-2} \\ \hline \boldsymbol{c}_{t-1} \\ \hline \boldsymbol{c}_{t} \\ \hline \boldsymbol{c}_{t+1} \\ \hline \vdots \end{array}\right] \end{array}$$

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

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

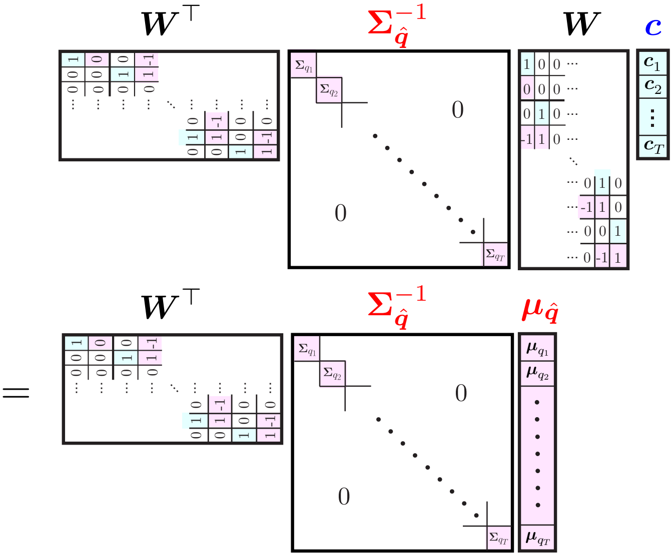
If state-output distribution is single Gaussian

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

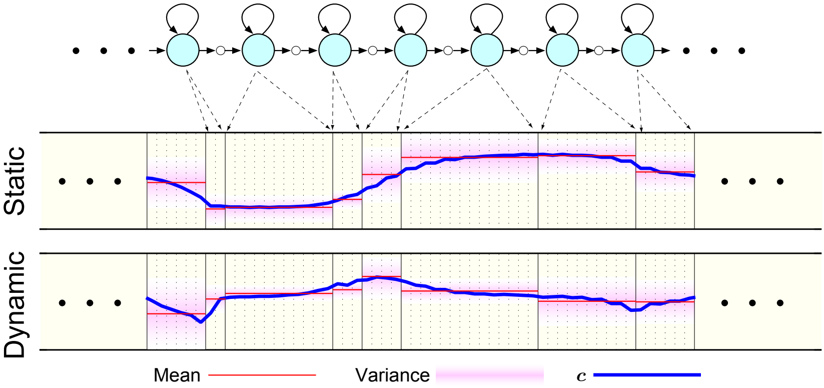
By setting $\partial \log \mathcal{N}(\boldsymbol{W}\boldsymbol{c}; \hat{\boldsymbol{\mu}}_{\hat{\boldsymbol{q}}}, \hat{\boldsymbol{\Sigma}}_{\hat{\boldsymbol{q}}}) / \partial \boldsymbol{c} = \boldsymbol{0}$

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

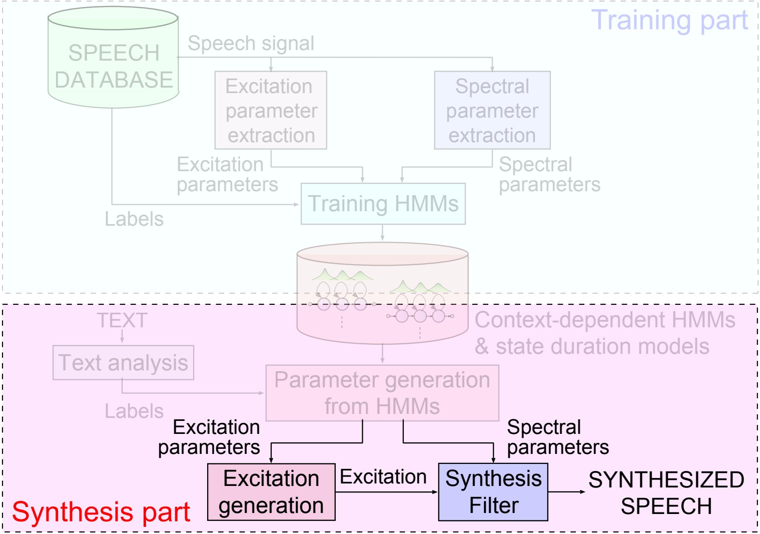
Speech parameter generation algorithm [9]



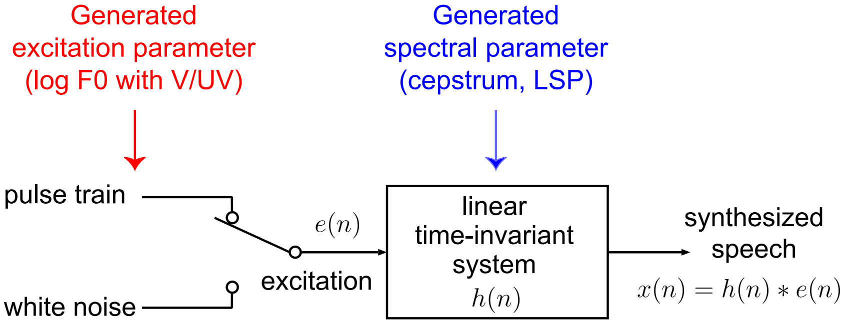
Generated speech parameter trajectory



HMM-based speech synthesis [4]



Waveform reconstruction



Generated
excitation parameter
(log F0 with V/UV)

Generated
spectral parameter
(cepstrum, LSP)

pulse train

white noise

$e(n)$

excitation

linear
time-invariant
system
$h(n)$

synthesized
speech

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

Synthesis filter

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

Characteristics of SPSS

- **Advantages**
  - Flexibility to change voice characteristics
    - Adaptation
    - Interpolation
  - Small footprint [10, 11]
  - Robustness [12]

- **Drawback**
  - Quality

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

Outline

Background

- HMM-based statistical parametric speech synthesis (SPSS)
- Flexibility
- Improvements

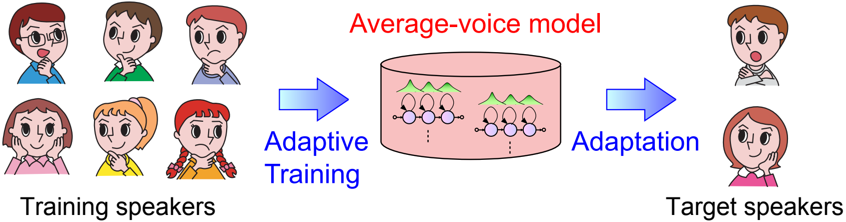
Statistical parametric speech synthesis with neural networks

- Deep neural network (DNN)-based SPSS
- Deep mixture density network (DMDN)-based SPSS
- Recurrent neural network (RNN)-based SPSS

Summary

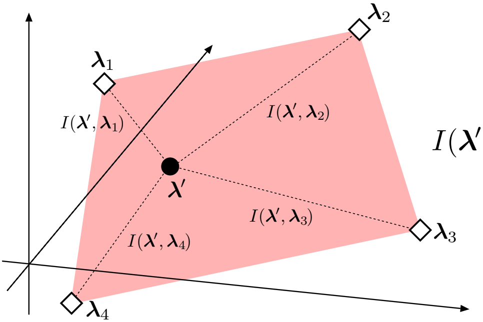
- Summary

Adaptation (mimicking voice) [13]



- 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

Interpolation (mixing voice) [14, 15, 16, 17]



$\boldsymbol{\lambda}$: 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
  → estimate representative HMM sets from data

Outline

Background

- HMM-based statistical parametric speech synthesis (SPSS)
- Flexibility
- Improvements

Statistical parametric speech synthesis with neural networks

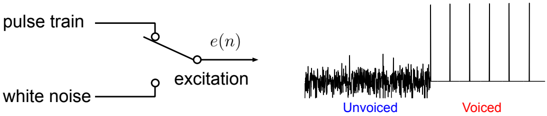
- Deep neural network (DNN)-based SPSS
- Deep mixture density network (DMDN)-based SPSS
- Recurrent neural network (RNN)-based SPSS

Summary

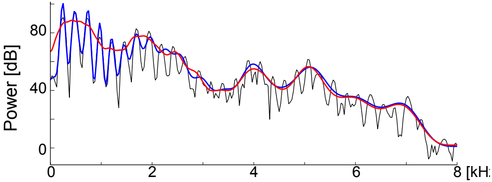
- Summary

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

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

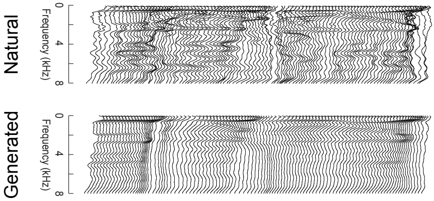
- **Piece-wise constatnt statistics** → Dynamical model
  - Trended HMM
  - Polynomial segment model
  - Trajectory HMM

- **Conditional independence assumption** → Graphical model
  - Buried Markov model
  - Autoregressive HMM
  - Trajectory HMM

- **Weak duration modeling** → Explicit duration model
  - Hidden semi-Markov model

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)

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)

Outline

Background
- HMM-based statistical parametric speech synthesis (SPSS)
- Flexibility
- Improvements

Statistical parametric speech synthesis with neural networks
- Deep neural network (DNN)-based SPSS
- Deep mixture density network (DMDN)-based SPSS
- Recurrent neural network (RNN)-based SPSS

Summary
- Summary

Linguistic → acoustic mapping

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

Linguistic → acoustic mapping

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

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

Linguistic → acoustic mapping

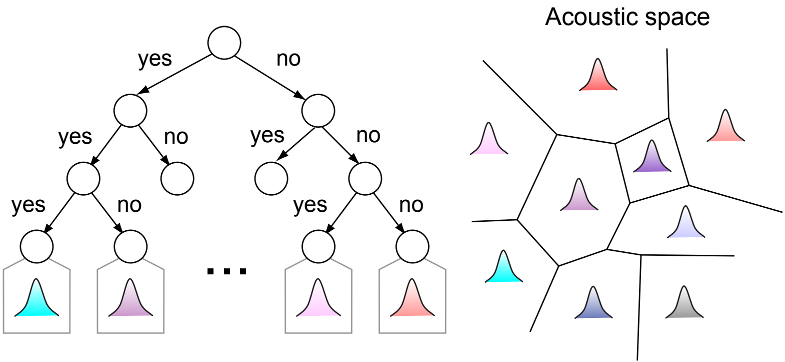
- **Training**
  Learn relationship between linguistc & acoustic features

- **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)

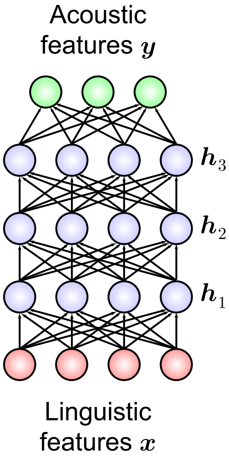
Effective modeling is essential

HMM-based acoustic modeling for SPSS [4]



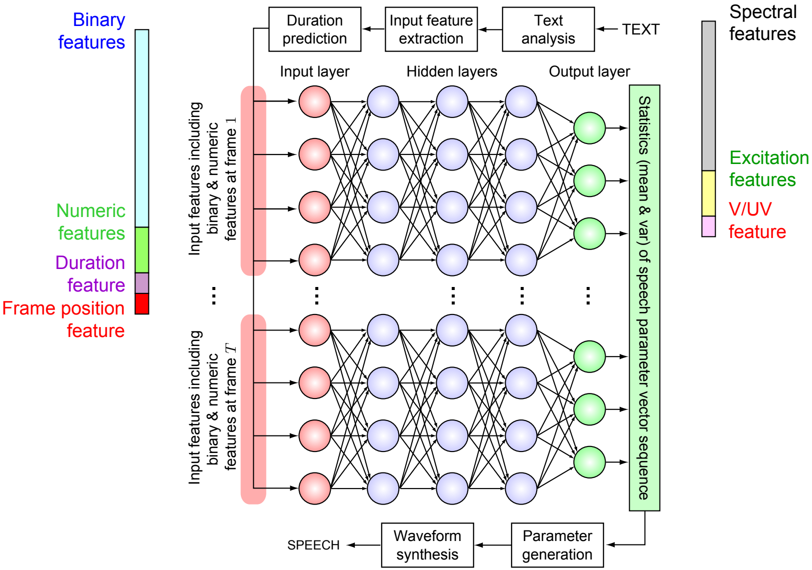
- Decision tree-clustered HMM with GMM state-output distributions

DNN-based acoustic modeling for SPSS [18]



- DNN represents conditional distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$
- DNN replaces decision trees and GMMs

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

Spectral features

Excitation features

V/UV feature

SPEECH

Waveform synthesis

Parameter generation

Advantages of NN-based acoustic modeling

- **Integrating feature extraction**
  - Can model high-dimensional, highly correlated features efficiently
  - Layered architecture w/ non-linear operations
    → Integrated feature extraction to acoustic modeling

Advantages of NN-based acoustic modeling

- **Integrating feature extraction**
  - Can model high-dimensional, highly correlated features efficiently
  - Layered architecture w/ non-linear operations
    → Integrated feature extraction to acoustic modeling
- **Distributed representation**
  - Can be exponentially more efficient than fragmented representation
  - Better representation ability with fewer parameters

Advantages of NN-based acoustic modeling

- **Integrating feature extraction**
  - Can model high-dimensional, highly correlated features efficiently
  - Layered architecture w/ non-linear operations
    → Integrated feature extraction to acoustic modeling
- **Distributed representation**
  - Can be exponentially more efficient than fragmented representation
  - Better representation ability with fewer parameters
- **Layered hierarchical structure in speech production**
  - concept → linguistic → articulatory → waveform

Framework

Is this new? . . . no

- NN [19]
- RNN [20]

Framework

Is this new? . . . no

- NN [19]
- RNN [20]

What's the difference?

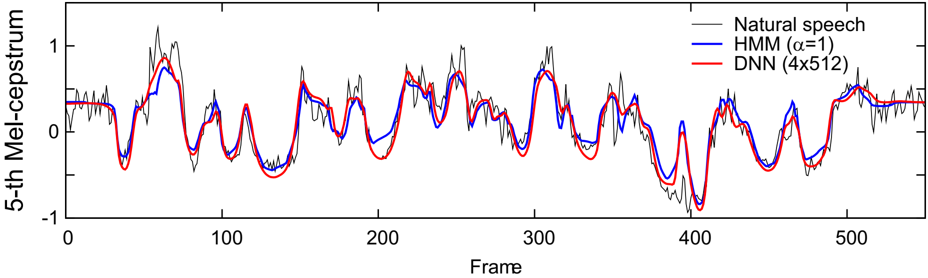
- More layers, data, computational resources
- Better learning algorithm
- Statistical parametric speech synthesis techniques

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 [21], MSD F_0 [22], MDL [23]
DNN architecture	1–5 layers, 256/512/1024/2048 units/layer sigmoid, continuous F_0 [24]
Postprocessing	Postfiltering in cepstrum domain [25]

Example of speech parameter trajectories

w/o grouping questions, numeric contexts, silence frames removed



Subjective evaluations

Compared HMM-based systems with DNN-based ones with similar # of parameters

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

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

Outline

Background

- HMM-based statistical parametric speech synthesis (SPSS)
- Flexibility
- Improvements

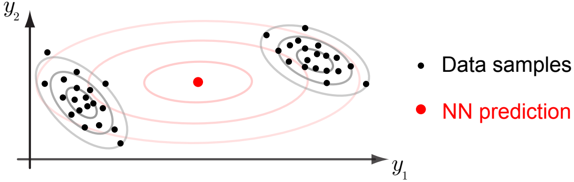
Statistical parametric speech synthesis with neural networks

- Deep neural network (DNN)-based SPSS
- Deep mixture density network (DMDN)-based SPSS
- Recurrent neural network (RNN)-based SPSS

Summary

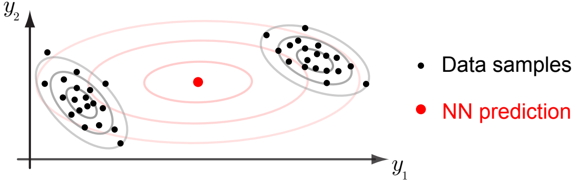
- Summary

Limitations of DNN-based acoustic modeling



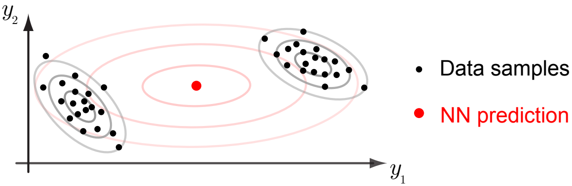
- **Unimodality**
  - Human can speak in different ways → one-to-many mapping
  - NN trained by MSE loss → approximates conditional mean

Limitations of DNN-based acoustic modeling



- **Unimodality**
  - Human can speak in different ways → 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

Limitations of DNN-based acoustic modeling

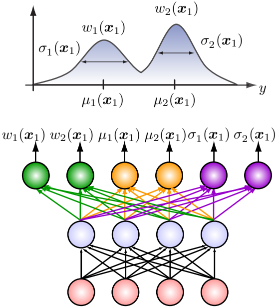


- **Unimodality**
  - Human can speak in different ways → one-to-many mapping
  - NN trained by MSE loss → approximates conditional mean

- **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 [26]

Mixture density network [26]



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

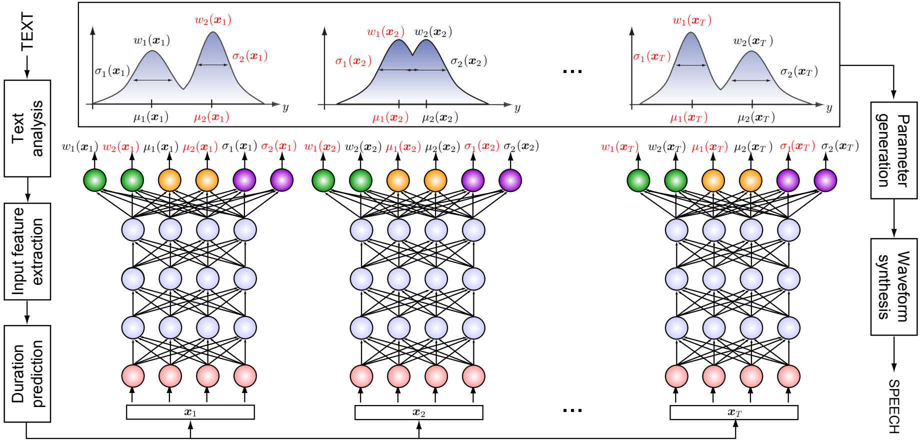
$$\mu_1(\boldsymbol{x}) = z_3 \qquad \mu_1(\boldsymbol{x}) = z_4$$

🟣 : 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 [27]



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 [28] (hidden) / Mixture density (output) 1–16 mix
Optimization	AdaDec [29] (variant of AdaGrad [30]) 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	**3.537 ± 0.113**
	2 mix	3.397 ± 0.115
DNN	4×1024	3.635 ± 0.127
	5×1024	**3.681 ± 0.109**
	6×1024	3.652 ± 0.108
	7×1024	3.637 ± 0.129
DMDN (4×1024)	1 mix	3.654 ± 0.117
	2 mix	3.796 ± 0.107
	4 mix	3.766 ± 0.113
	8 mix	**3.805 ± 0.113**
	16 mix	3.791 ± 0.102

Outline

Background

- HMM-based statistical parametric speech synthesis (SPSS)
- Flexibility
- Improvements

Statistical parametric speech synthesis with neural networks

- Deep neural network (DNN)-based SPSS
- Deep mixture density network (DMDN)-based SPSS
- Recurrent neural network (RNN)-based SPSS

Summary

- Summary

Limitations of DNN/DMDN-based acoustic modeling

- **Fixed time span for input features**
  - Fixed number of preceding / succeeding contexts (e.g., ± 2 phonemes/syllable stress) are used as inputs
  - Difficult to incorporate long time span contextual effect
- **Frame-by-frame mapping**
  - Each frame is mapped independently
  - Smoothing using dynamic feature constraints is still essential

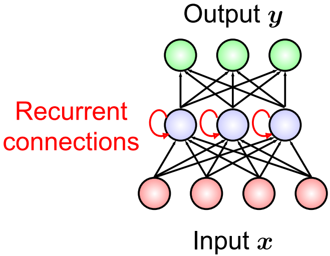
Limitations of DNN/DMDN-based acoustic modeling

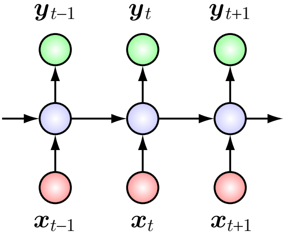
- **Fixed time span for input features**
  - Fixed number of preceding / succeeding contexts (e.g., ± 2 phonemes/syllable stress) are used as inputs
  - Difficult to incorporate long time span contextual effect

- **Frame-by-frame mapping**
  - Each frame is mapped independently
  - Smoothing using dynamic feature constraints is still essential

Recurrent connections → Recurrent NN (RNN) [31]

Basic RNN





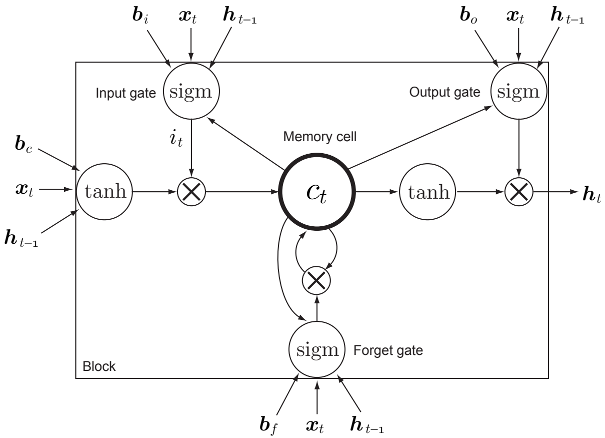
- **Only able to use previous contexts**
  → bidirectional RNN [31]

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

  → long short-term memory (LSTM) RNN [32]

Long short-term memory (LSTM) [32]

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

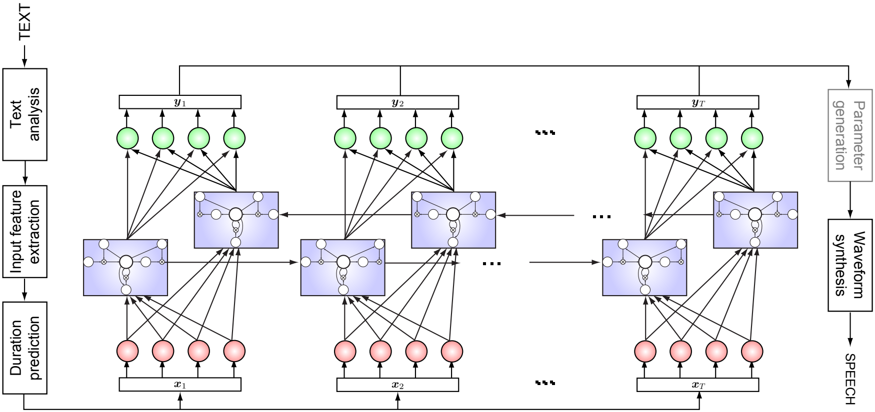


Input gate: Write

Output gate: Read

Forget gate: Reset

LSTM-based SPSS [33, 34]



Experimental setup

Database	US English female speaker
Train / dev set data	34632 & 100 sentences
Sampling rate	16 kHz
Analysis window	25-ms width / 5-ms shift
Linguistic features	DNN: 449 LSTM: 289
Acoustic features	0–39 mel-cepstrum $\log F_0$, 5-band aperiodicity (Δ, Δ^2)
DNN	4 hidden layers, 1024 units/hidden layer ReLU (hidden) / Linear (output) AdaDec [29] on GPU
LSTM	1 forward LSTM layer 256 units, 128 projection Asynchronous SGD on CPUs [35]
Postprocessing	Postfiltering in cepstrum domain [25]

Subjective evaluations

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

DNN		LSTM			Stats	
w/ Δ	w/o Δ	w/ Δ	w/o Δ	Neutral	z	p
50.0	14.2	–	–	35.8	12.0	$< 10^{-10}$
–	–	**30.2**	15.6	54.2	5.1	$< 10^{-6}$
15.8	–	**34.0**	–	50.2	-6.2	$< 10^{-9}$
28.4	–	–	33.6	38.0	-1.5	0.138

Samples

- DNN (w/o dynamic features)

  🔊 🔊 🔊 🔊 🔊

- DNN (w/ dynamic features)

  🔊 🔊 🔊 🔊 🔊

- LSTM (w/o dynamic features)

  🔊 🔊 🔊 🔊 🔊

- LSTM (w/ dynamic features)

  🔊 🔊 🔊 🔊 🔊

Outline

Background

- HMM-based statistical parametric speech synthesis (SPSS)
- Flexibility
- Improvements

Statistical parametric speech synthesis with neural networks

- Deep neural network (DNN)-based SPSS
- Deep mixture density network (DMDN)-based SPSS
- Recurrent neural network (RNN)-based SPSS

Summary

- Summary

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)

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] Y. Morioka, S. Kataoka, H. Zen, Y. Nankaku, K. Tokuda, and T. Kitamura.
Miniaturization of HMM-based speech synthesis.
In *Proc. Autumn Meeting of ASJ*, pages 325–326, 2004.
(in Japanese).

[11] S.-J. Kim, J.-J. Kim, and M.-S. Hahn.
HMM-based Korean speech synthesis system for hand-held devices.
IEEE Trans. Consum. Electron., 52(4):1384–1390, 2006.

[12] J. Yamagishi, Z.H. Ling, and S. King.
Robustness of HMM-based speech synthesis.
In *Proc. Interspeech*, pages 581–584, 2008.

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

[14] 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.

[15] 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.

[16] 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.

References III

[17] 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.

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

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

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

[21] 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.

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

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

[24] 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.

References IV

[25] 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.

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

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

[28] 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.

[29] 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.

[30] 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.

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

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

References V

[33] 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/`.

[34] H. Zen, H. Sak, A. Graves, and A. Senior.
Statistical parametric speech synthesis using recurrent neural networks.
In *UKSpeech Conference*, 2014.

[35] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng.
Large scale distributed deep networks.
In *Proc. NIPS*, 2012.