Acoustic Modeling for Speech Synthesis

Heiga Zen

June 3rd, 2016@Nitech

Outline

Background

HMM-based acoustic modeling
- Training & synthesis
- Limitations

ANN-based acoustic modeling
- Feedforward NN
- RNN

Conclusion

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

Automatic speech recognition (ASR)

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

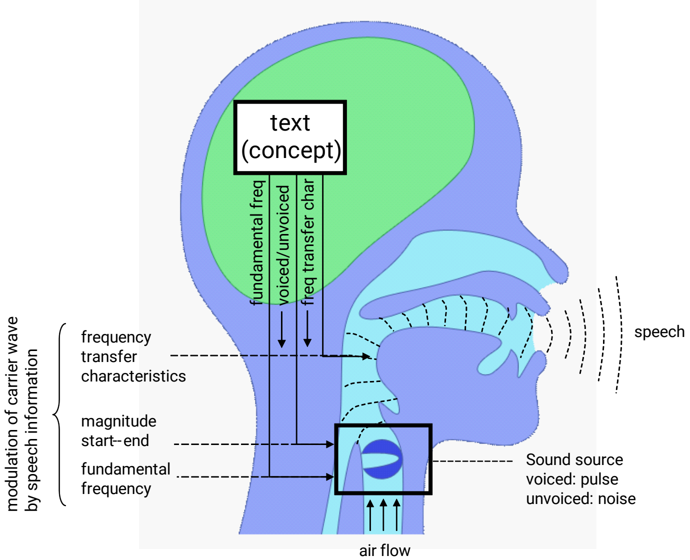
Statistical machine translation (SMT)

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

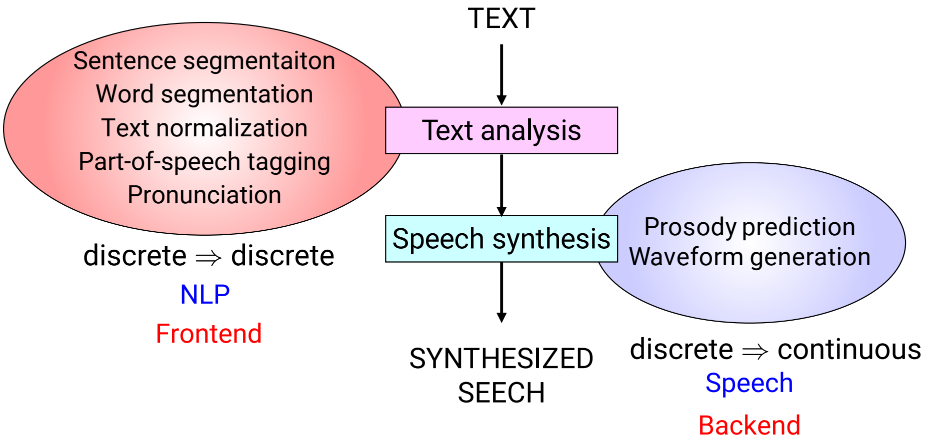
Text-to-speech synthesis (TTS)

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

Speech production process

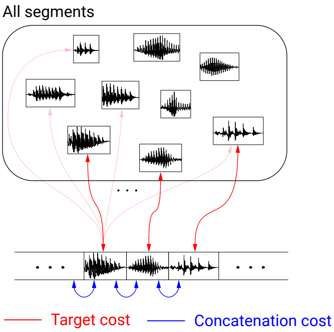


Typical flow of TTS system



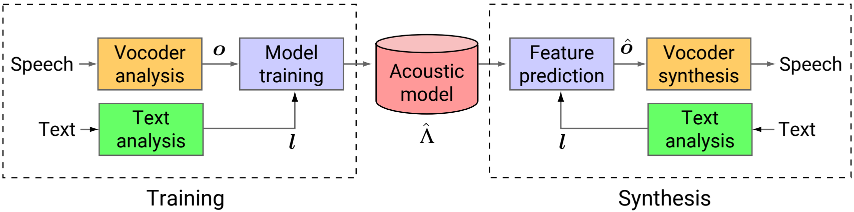
This presentation mainly talks about backend

Concatenative speech synthesis



- Concatenate actual small speech segments from database
  $\rightarrow$ Very high segmental naturalness
- Single segment per unit (e.g., diphone) $\rightarrow$ diphone synthesis [1]
- Multiple segments per unit $\rightarrow$ unit selection synthesis [2]

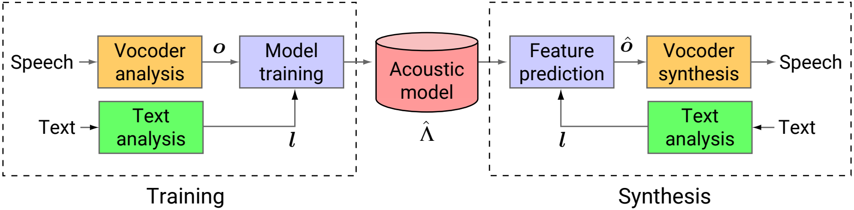
Statistical parametric speech synthesis (SPSS) [4]



- Parametric representation rather than waveform
- Model relationship between linguistic & acoustic features
- Predict acoustic features then reconstruct waveform

SPSS can use any acoustic model, but HMM-based one is very popular
→ HMM-based speech synthesis [3]

Statistical parametric speech synthesis (SPSS) [4]



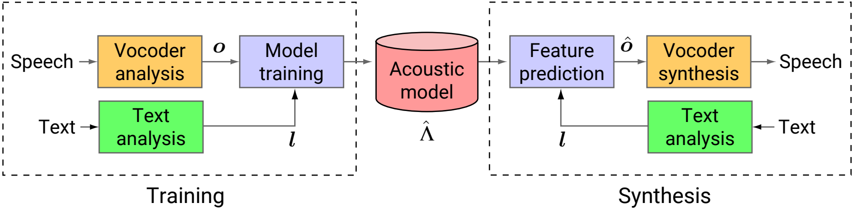
Pros

- Small footprint
- Flexibility to change voice characteristics
- Robust to data sparsity and noise/mistakes in data

Cons

- Segmental naturalness

Major factors for naturalness degradation



- **Vocoder analysis/synthesis**
  - *How to parameterize speech?*
- **Acoustic model**
  - *How to represent relationship between speech & text?*
- **Oversmoothing**
  - *How to generate speech from model?*

Formulation of SPSS

Training

- Extract linguistic features $\boldsymbol{l}$ & acoustic features $\boldsymbol{o}$
- Train acoustic model Λ given $(\boldsymbol{o}, \boldsymbol{l})$

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

Synthesis

- Extract $\boldsymbol{l}$ from text to be synthesized
- Generate most probable $\boldsymbol{o}$ from $\hat{\Lambda}$ then reconstruct waveform

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

Formulation of SPSS

Training

- Extract linguistic features $\boldsymbol{l}$ & acoustic features $\boldsymbol{o}$
- Train acoustic model Λ given $(\boldsymbol{o}, \boldsymbol{l})$

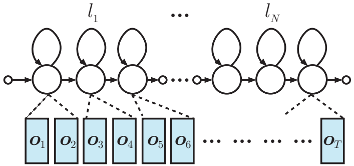
$$\hat{\Lambda} = \arg\max_{\Lambda} p(\boldsymbol{o} \mid \boldsymbol{l}, \Lambda)$$

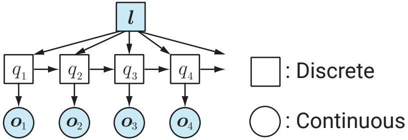
Synthesis

- Extract $\boldsymbol{l}$ from text to be synthesized
- Generate most probable $\boldsymbol{o}$ from $\hat{\Lambda}$ then reconstruct waveform

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

Training – HMM-based acoustic modeling





$$p(\boldsymbol{o} \mid \boldsymbol{l}, \Lambda) = \sum_{\forall \boldsymbol{q}} p(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda) P(\boldsymbol{q} \mid \boldsymbol{l}, \Lambda)$$

$\boldsymbol{q}$: hidden states

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

q_t: hidden state at t

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

ML estimation of HMM parameters $\rightarrow$ Baum-Welch (EM) algorithm [5]

Training – Linguistic features

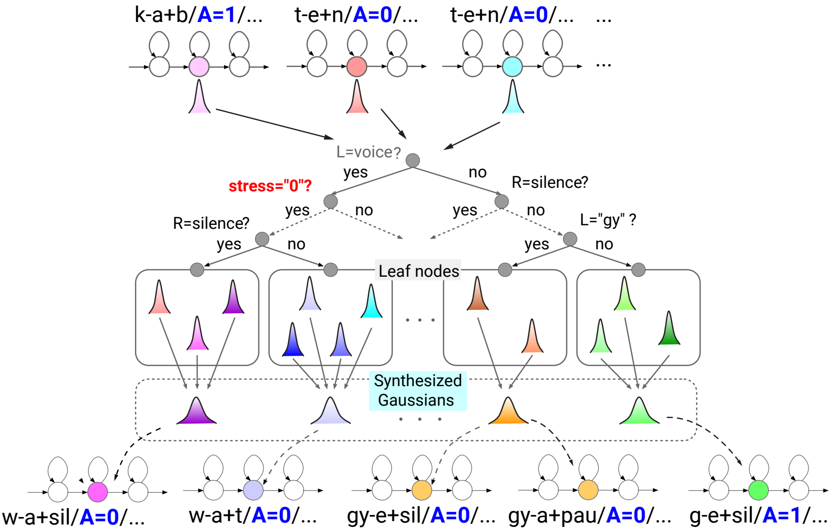
Linguistic features: phonetic, grammatical, & prosodic features

- **Phoneme**
  phoneme identity, position
- **Syllable**
  length, accent, stress, tone, vowel, position
- **Word**
  length, POS, grammar, prominence, emphasis, position, pitch accent
- **Phrase**
  length, type, position, intonation
- **Sentence**
  length, type, position

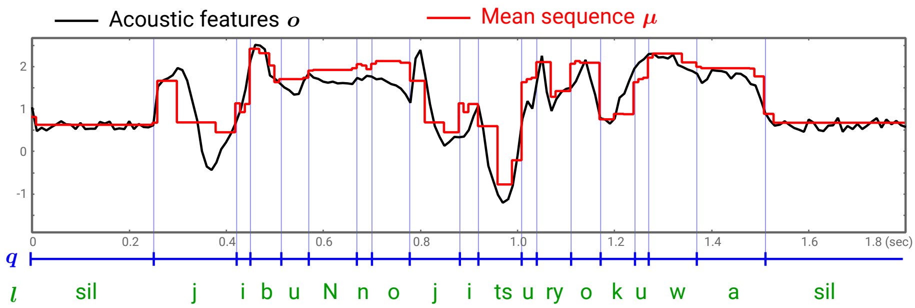
...

$\rightarrow$ Impossible to have enough data to cover all combinations

Training – ML decision tree-based state clustering [6]



Training – Example



Formulation of SPSS

Training

- Extract linguistic features l & acoustic features o
- Train acoustic model Λ given $(\boldsymbol{o}, \boldsymbol{l})$

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

Synthesis

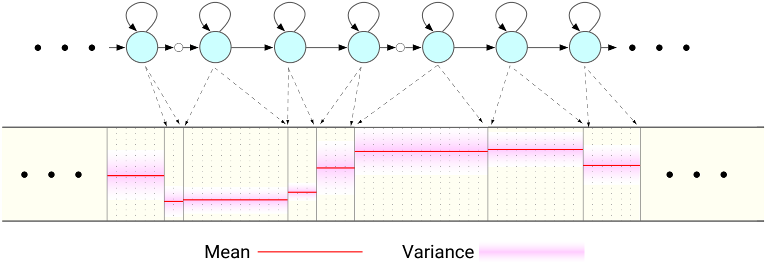
- Extract l from text to be synthesized
- Generate most probable o from $\hat{\Lambda}$ then reconstruct waveform

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

Synthesis – Predict most probable acoustic features

$$\begin{aligned}
\hat{o} &= \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \boldsymbol{l}, \hat{\Lambda}) \\
&= \arg\max_{\boldsymbol{o}} \sum_{\forall \boldsymbol{q}} p(\boldsymbol{o}, \boldsymbol{q} \mid \boldsymbol{l}, \hat{\Lambda}) \\
&\approx \arg\max_{\boldsymbol{o}} \max_{\boldsymbol{q}} p(\boldsymbol{o}, \boldsymbol{q} \mid \boldsymbol{l}, \hat{\Lambda}) \\
&= \arg\max_{\boldsymbol{o}} \max_{\boldsymbol{q}} p(\boldsymbol{o} \mid \boldsymbol{q}, \hat{\Lambda}) P(\boldsymbol{q} \mid \boldsymbol{l}, \hat{\Lambda}) \\
&\approx \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{q}}, \hat{\Lambda}) \quad s.t. \quad \hat{\boldsymbol{q}} = \arg\max_{\boldsymbol{q}} P(\boldsymbol{q} \mid \boldsymbol{l}, \hat{\Lambda}) \\
&= \arg\max_{\boldsymbol{o}} \mathcal{N}\left(\boldsymbol{o}; \boldsymbol{\mu}_{\hat{\boldsymbol{q}}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}\right) \\
&= \boldsymbol{\mu}_{\hat{\boldsymbol{q}}} \\
&= \left[\boldsymbol{\mu}_{\hat{q}_1}^{\top}, \ldots, \boldsymbol{\mu}_{\hat{q}_T}^{\top}\right]^{\top}
\end{aligned}$$

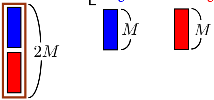
Synthesis – Most probable acoustic features given HMM

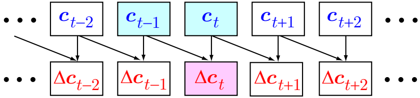


$\hat{o} \rightarrow$ step-wise $\rightarrow$ discontinuity can be perceived

Synthesis – Using dynamic feature constraints [7]

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





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

Synthesis – Speech parameter generation algorithm [7]

$$\hat{o} = \arg\max_{\boldsymbol{o}} p(\boldsymbol{o} \mid \hat{\boldsymbol{q}}, \hat{\Lambda}) \quad s.t. \quad \boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$$

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

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

$$\frac{\partial}{\partial \boldsymbol{c}} \log \mathcal{N}(\boldsymbol{W}\boldsymbol{c}; \boldsymbol{\mu}_{\hat{\boldsymbol{q}}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}) \propto \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}^{-1} \boldsymbol{W}\boldsymbol{c} - \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}^{-1} \boldsymbol{\mu}_{\hat{\boldsymbol{q}}}$$

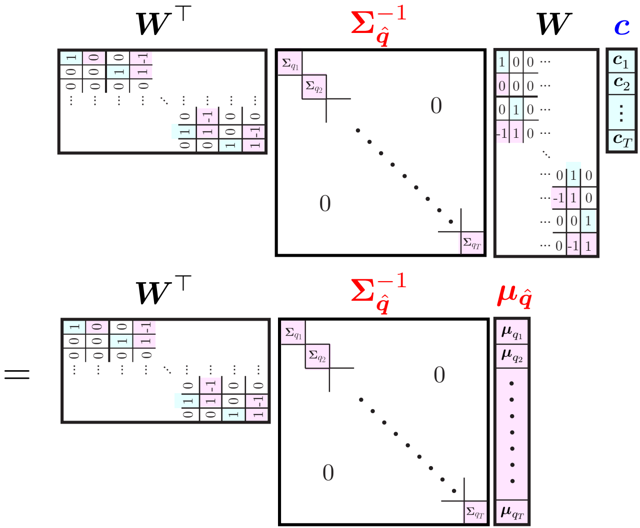
$$\boldsymbol{W}^\top \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}^{-1} \boldsymbol{W}\boldsymbol{c} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}^{-1} \boldsymbol{\mu}_{\hat{\boldsymbol{q}}}$$

where

$$\boldsymbol{\mu}_{\boldsymbol{q}} = \left[\boldsymbol{\mu}_{q_1}^\top, \boldsymbol{\mu}_{q_2}^\top, \ldots, \boldsymbol{\mu}_{q_T}^\top\right]^\top$$

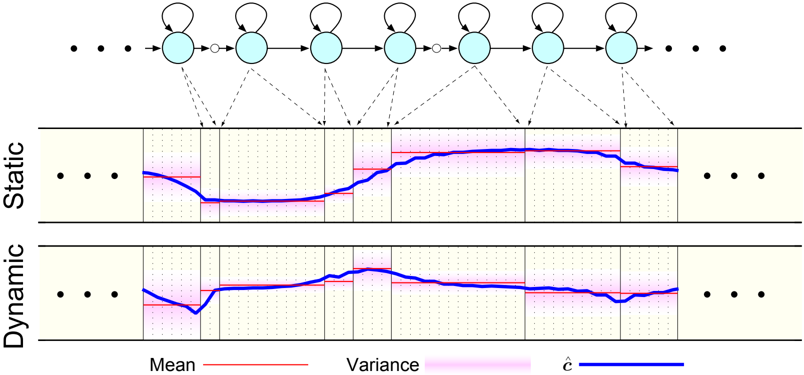
$$\boldsymbol{\Sigma}_{\boldsymbol{q}} = \mathrm{diag}\left[\boldsymbol{\Sigma}_{q_1}, \boldsymbol{\Sigma}_{q_2}, \ldots, \boldsymbol{\Sigma}_{q_T}\right]$$

Synthesis – Speech parameter generation algorithm [7]



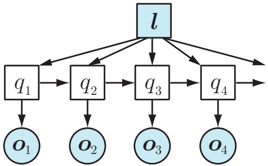
Synthesis – Most probable acoustic features

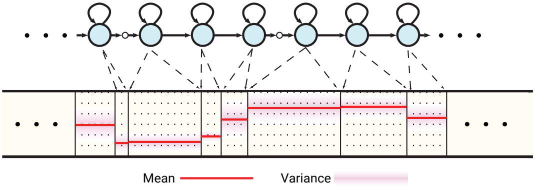
under constraints between static & dynamic features



HMM-based acoustic model – Limitations (1)

Stepwise statistics

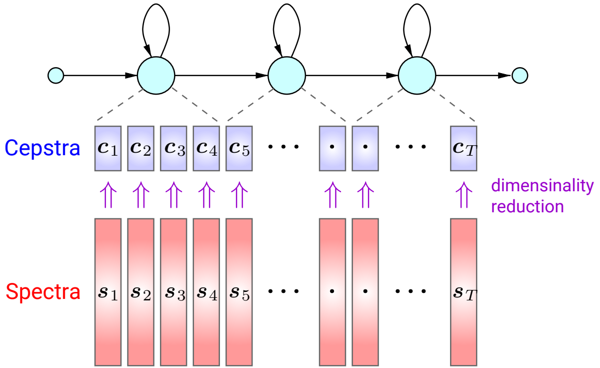




- Output probability only depends on the current state
- Within the same state, statistics are constant
  $\rightarrow$ Step-wise statistics
- Using dynamic feature constraints
  $\rightarrow$ Ad hoc & introduces inconsistency betw. training & synthesis [8]

HMM-based acoustic model – Limitations (2)

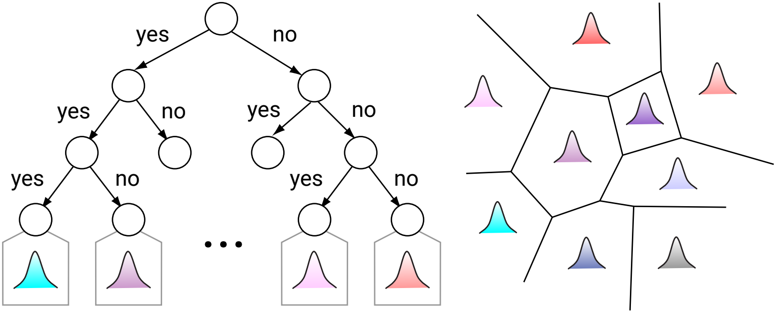
Difficulty to integrate feature extraction & modeling



- Spectra or waveforms are high-dimensional & highly correlated
- Hard to be modeled by HMMs with Gaussian + digonal covariance
  → Use low dimensional approximation (e.g., cepstra, LSPs)

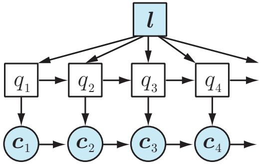
HMM-based acoustic model – Limitations (3)

Data fragmentation

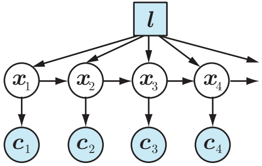


- Trees split input into clusters & put representative distributions
  $\rightarrow$ Inefficient to represent dependency betw. ling. & acoust. feats.
- Minor features are never used (e.g., word-level emphasis [9])
  $\rightarrow$ Little or no effect

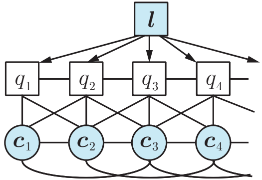
Alternatives – Stepwise statistics



ARHMM



LDM

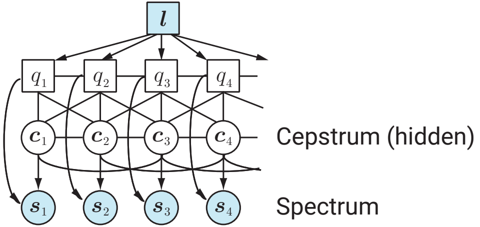


Trajectory HMM

- Autoregressive HMMs (ARHMMs) [10]
- Linear dynamical models (LDMs) [11, 12]
- Trajectory HMMs [8]
- $\cdots$

Most of them use clustering $\rightarrow$ Data fragmentation
Often employ trees from HMM $\rightarrow$ Sub-optimal

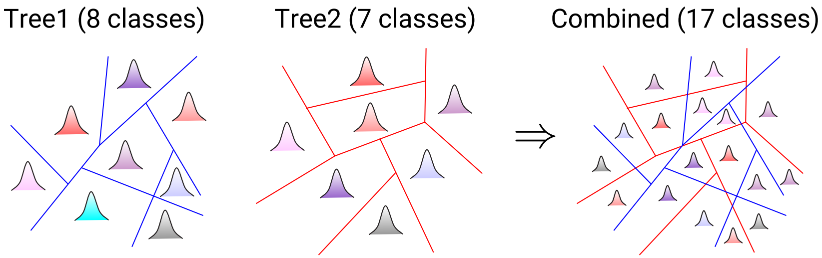
Alternatives – Difficulty to integrate feature extraction



- Statistical vocoder [13]
- Minimum generation error with log spectral distortion [14]
- Waveform-level model [15]
- Mel-cepstral analysis-integrated HMM [16]

Use clustering to build tying structure $\rightarrow$ Data fragmentation
Often employ trees from HMM $\rightarrow$ Sub-optimal

Alternatives – Data fragmentation



- Factorized decision tree [9, 17]
- Product of experts [18]

Each tree/expert still has data fragmentation → Data fragmentation
Fix other trees while building one tree [19, 20] → Sub-optimal

Linguistic $\rightarrow$ Acoustic mapping

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

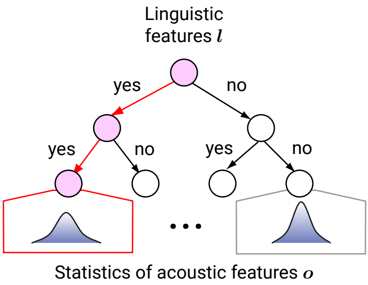
- **Synthesis**
  Map linguistic features to acoustic ones

- **Linguistic features used in SPSS**
  - Phoneme, syllable, word, phrase, utterance-level features
  - Around 50 different types
  - Sparse & correlated

Effective modeling is essential

Decision tree-based acoustic model

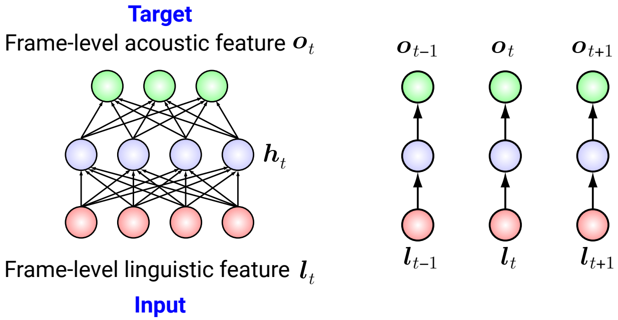
HMM-based acoustic model & alternatives
→ Actually decision tree-based acoustic model



Regression tree: linguistic features → Stats. of acoustic features

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

ANN-based acoustic model [21] – Overview



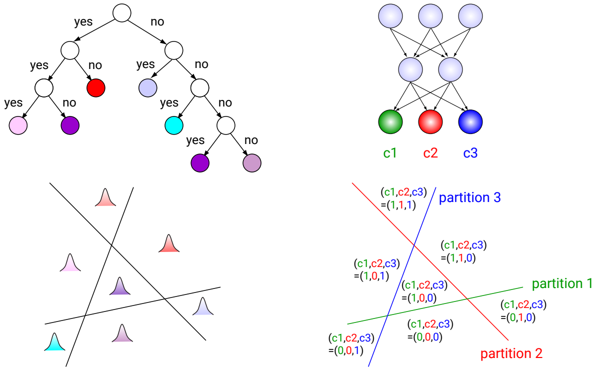
$$\boldsymbol{h}_t = f\left(\boldsymbol{W}_{hl}\boldsymbol{l}_t + \boldsymbol{b}_h\right) \qquad \hat{\boldsymbol{o}}_t = \boldsymbol{W}_{oh}\boldsymbol{h}_t + \boldsymbol{b}_o$$

$$\hat{\Lambda} = \arg\min_{\Lambda} \sum_t \|\boldsymbol{o}_t - \hat{\boldsymbol{o}}_t\|_2 \qquad \Lambda = \{\boldsymbol{W}_{hl}, \boldsymbol{W}_{oh}, \boldsymbol{b}_h, \boldsymbol{b}_o\}$$

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

ANN-based acoustic model [21] – Motivation (1)

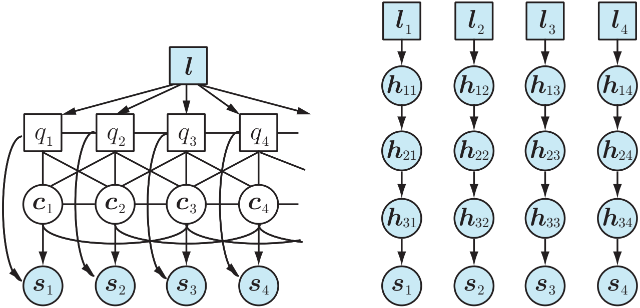
Distributed representation [22, 23]



- Fragmented: n terminal nodes $\rightarrow$ n classes (linear)
- Distributed: n binary units $\rightarrow$ 2^n classes (exponential)
- Minor features (e.g., word-level emphasis) can affect synthesis

ANN-based acoustic model [21] – Motivation (2)

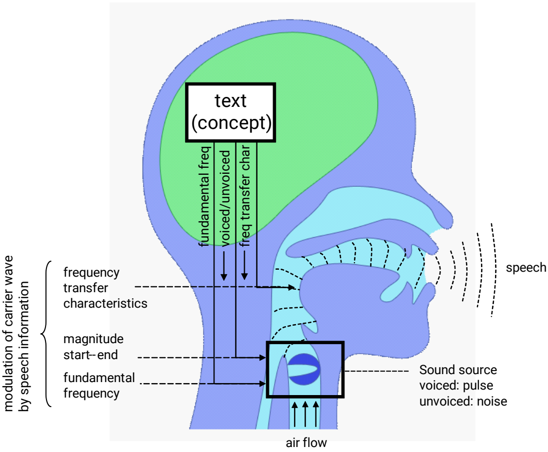
Integrate feature extraction [24, 25, 26]

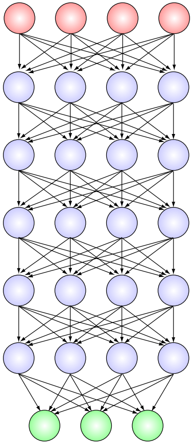


- Layered architecture with non-linear operations
- Can model high-dimensional/correlated linguistic/acoustic features
  $\rightarrow$ Feature extraction can be embedded in model itself

ANN-based acoustic model [21] – Motivation (3)

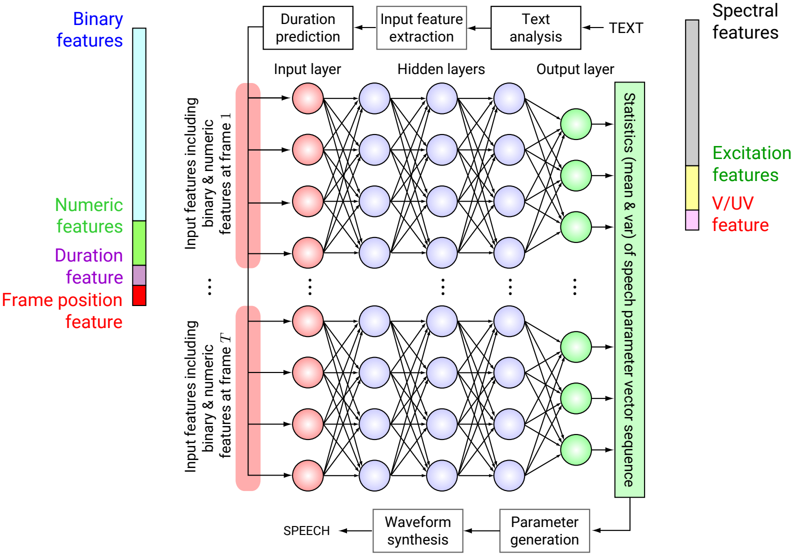
Implicitly mimic layered hierarchical structure in speech production



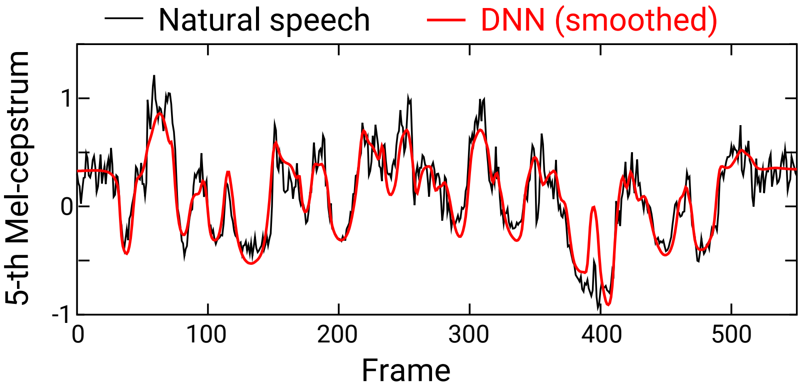


Concept → Linguistic → Articulator → Vocal tract → Waveform

DNN-based speech synthesis [21] – Implementation



DNN-based speech synthesis [21] – Example



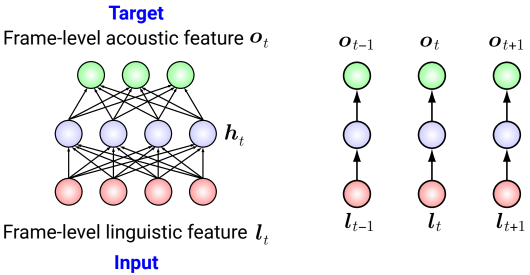
DNN-based speech synthesis [21] – Subjective eval.

Compared HMM- & DNN-based TTS w/ similar # of parameters

- US English, professional speaker, 30 hours of speech data
- Preference test
- 173 test sentences, 5 subjects per pair
- Up to 30 pairs per subject
- Crowd-sourced

Preference scores (%)			
HMM	DNN	No pref.	#layers $\times$ #units
15.8	**38.5**	45.7	4×256
16.1	**27.2**	56.7	4×512
12.7	**36.6**	50.7	4×1024

Feedforward NN-based acoustic model – Limitation

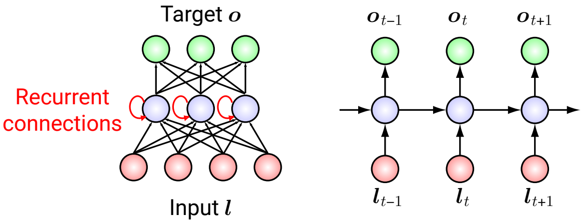


Each frame is mapped independently $\rightarrow$ Smoothing is still essential

Preference scores (%)		
DNN with dyn	DNN without dyn	No pref.
67.8	12.0	20.0

Recurrent connections $\rightarrow$ Recurrent NN (RNN) [27]

RNN-based acoustic model [28, 29]

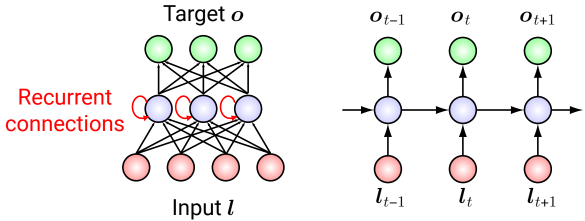


$$\boldsymbol{h}_t = f\left(\boldsymbol{W}_{hl}\boldsymbol{l}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h\right) \quad \hat{\boldsymbol{o}}_t = \boldsymbol{W}_{oh}\boldsymbol{h}_t + \boldsymbol{b}_o$$

$$\hat{\Lambda} = \arg\min_{\Lambda} \sum_t \|\boldsymbol{o}_t - \hat{\boldsymbol{o}}_t\|_2 \quad \Lambda = \{\boldsymbol{W}_{hl}, \boldsymbol{W}_{hh}, \boldsymbol{W}_{oh}, \boldsymbol{b}_h, \boldsymbol{b}_o\}$$

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

RNN-based acoustic model [28, 29]



- **Only able to use previous contexts**
  $\rightarrow$ Bidirectional RNN [27]: $\hat{\boldsymbol{o}}_t \approx \mathbb{E}\left[\boldsymbol{o}_t \mid \boldsymbol{l}_1, \ldots, \boldsymbol{l}_T\right]$
- **Trouble accessing long-range contexts**
  - Information in hidden layers loops quickly decays over time
  - Prone to being overwritten by new information from inputs

  $\rightarrow$ Long short-term memory (LSTM) [30]

LSTM-RNN-based acoustic model [29]

Subjective preference test (same US English data)

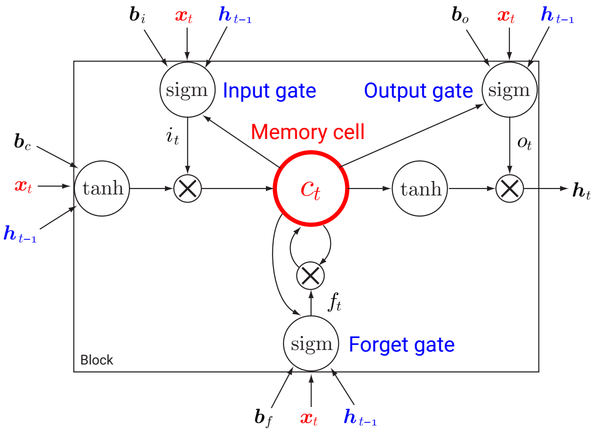
DNN: 3 layers, 1024 units
LSTM: 1 layer, 256 LSTM units

DNN with dyn	LSTM with dyn	No pref.
18.4	**34.9**	47.6

LSTM with dyn	LSTM without dyn	No pref.
21.0	12.2	66.8

→ Smoothing was still effective

Why?



Gate output: 0 -- 1

Input gate == 1
→ Write memory

Forget gate == 0
→ Reset memory

Output gate == 1
→ Read memory

- Gates in LSTM units: 0/1 switch controlling information flow
- Can produce rapid change in outputs
  → Discontinuity

How?

- Using loss function incorporating continuity
- Integrate smoothing → Recurrent output layer [29]

$$\boldsymbol{h}_t = \text{LSTM}\left(\boldsymbol{l}_t\right) \qquad \hat{\boldsymbol{o}}_t = \boldsymbol{W}_{oh}\boldsymbol{h}_t + \boldsymbol{W}_{oo}\hat{\boldsymbol{o}}_{t-1} + \boldsymbol{b}_o$$

Works pretty well

LSTM with dyn (Feedforward)	LSTM without dyn (Recurrent)	No pref.
21.8	21.0	57.2

Having two smoothing togeter doesn't work well → Oversmoothing?

LSTM with dyn (Recurrent)	LSTM without dyn (Recurrent)	No pref.
16.6	**29.2**	54.2

Low-latency TTS by unidirectional LSTM-RNN [29]

HMM / DNN

- Smoothing by dyn. needs to solve set of T linear equations

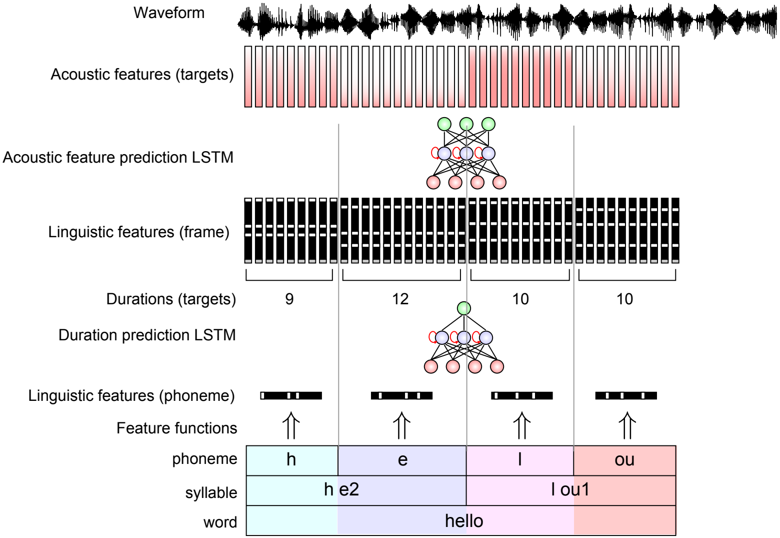
$$\boldsymbol{W}^\top \boldsymbol{\Sigma}_{\hat{q}}^{-1} \boldsymbol{W} \boldsymbol{c} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\hat{q}}^{-1} \boldsymbol{\mu}_{\hat{q}} \qquad T\text{: Utterance length}$$

- Order of operations to determine the first frame c_1 (latency)
  - Cholesky decomposition [7] $\rightarrow \mathcal{O}(T)$
  - Recursive approximation [31] $\rightarrow \mathcal{O}(L)$ $\quad L$: lookahead, $10 \sim 30$

Unidirectional LSTM with recurrent output layer [29]

- No smoothing required, fully time-synchronous w/o lookahead
- Order of latency $\rightarrow \mathcal{O}(1)$

Low-latency TTS by LSTM-RNN [29] – Implementation



Some comments

Is this new? . . . **no**

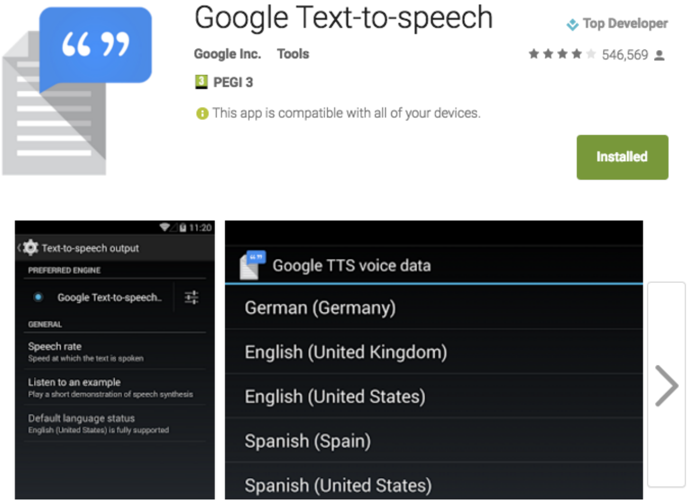
- Feedforward NN-based speech synthesis [32]
- RNN-based speech synthesis [33]

What's the difference?

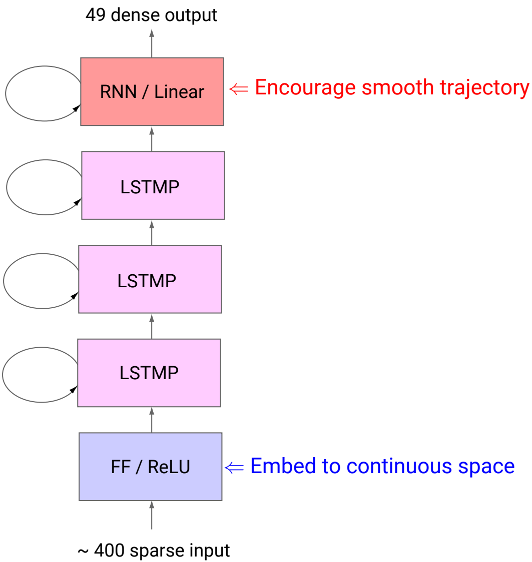
- More layers, data, computational resources
- Better learning algorithm
- Modern SPSS techniques

Making LSTM-RNN-based TTS into production

Client-side (local) TTS for Android



Network architecture



Further optimization

- **Disk footprint**
  HMM $\rightarrow$ 8-bit quantized [34]
  RNN $\rightarrow$ Float
  $\rightarrow$ **Weight quantization**

- **Computational cost at inference**
  HMM $\rightarrow$ Traversing decision trees (state) + parameter generation
  RNN $\rightarrow$ Matrix-Vector multiplication (frame)
  $\rightarrow$ **Multi-frame inference**

- **Robustness**
  HMM $\rightarrow$ "Soft" alignments using the Baum-Welch algorithm
  RNN $\rightarrow$ Typically relies on fixed alignments [21]
  $\rightarrow$ ϵ**-contaminated Gaussian loss function**

Weight quantization

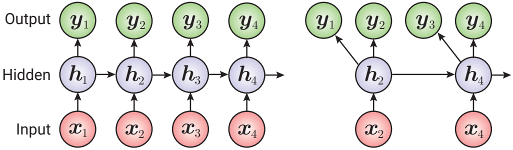
8-bit quantization of ANN weights to reduce footprint [35]

Language	Preference scores (%)		
	`int8`	`float`	No pref.
English (GB)	13.0	12.2	74.8
English (NA)	8.0	10.0	82.0
French	4.7	3.8	91.5
German	12.5	8.8	78.7
Italian	12.0	9.8	78.2
Spanish (ES)	8.8	7.5	83.7

No degradation by weight quantization

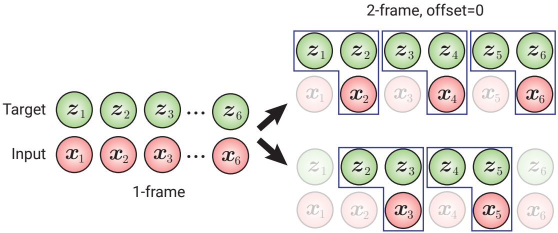
Multi-frame inference

Bundle multiple targets to a single one [36]



(a) 1-frame (b) 2-frame

Data augmentation



Multi-frame inference

4-frame inference w/ data augmentation

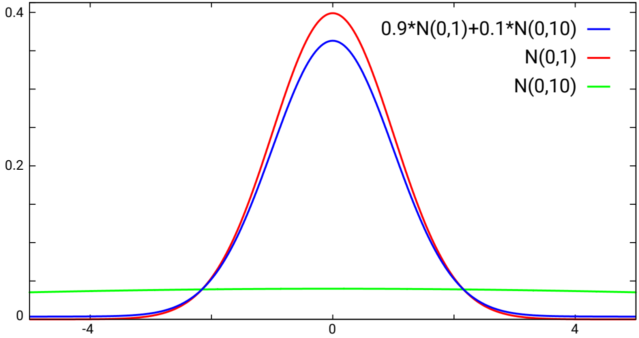
	Preference scores (%)		
Language	`4-frame+`	`1-frame`	No pref.
English (GB)	25.7	20.2	54.2
English (NA)	8.5	6.2	85.3
French	18.8	18.6	62.6
German	19.3	22.2	58.5
Italian	13.5	14.4	72.1
Spanish (ES)	12.8	17.0	70.3

No degradation by multi-frame inference

ϵ-contaminated Gaussian loss

Use heavier-tailed distribution as loss

$$\mathcal{L}(\boldsymbol{z};\boldsymbol{x},\boldsymbol{\Lambda}) = -\log\left\{(1-\epsilon)\mathcal{N}\left(\boldsymbol{z}; f(\boldsymbol{x};\boldsymbol{\Lambda}),\boldsymbol{\Sigma}\right) + \epsilon\mathcal{N}\left(\boldsymbol{z}; f(\boldsymbol{x};\boldsymbol{\Lambda}), c\boldsymbol{\Sigma}\right)\right\}$$



ϵ-contaminated Gaussian loss

	Preference scores (%)		
Language	CG	L2	No pref.
English (GB)	**27.4**	18.1	54.5
English (NA)	7.6	6.8	85.6
French	**24.6**	15.9	59.5
German	17.1	20.8	62.1
Italian	**16.0**	10.6	73.4
Spanish (ES)	16.0	13.4	70.6

Comparison w/ HMM-based SPSS

- HMMs & LSTM-RNNs were quantized into 8-bit integers
- Same training data & text processing front-end
- Average disk footprint; HMM: 1,560KB LSTM-RNN: 454.5KB
- HMM: Time-recursive parameter generation [31] w/ 10-frame delay

Length	Latency (ms)		Total (ms)	
	LSTM	HMM	LSTM	HMM
character	12.5	19.5	49.8	49.6
word	14.6	25.3	61.2	80.5
sentence	31.4	55.4	257.3	286.2
paragraph	64.1	117.7	2216.1	2400.8

Comparison w/ HMM-based SPSS

	Preference scores (%)		
Language	LSTM	HMM	No pref.
English (GB)	31.6	28.1	40.3
English (NA)	**30.6**	15.9	53.5
French	**68.6**	8.4	23.0
German	**52.8**	19.3	27.9
Italian	**84.8**	2.9	12.3
Spanish (ES)	**72.6**	10.6	16.8

Comparison w/ concatenative TTS

Language	LSTM	Hybrid	No pref.
Arabic	13.9	**22.1**	64.0
Cantonese	**25.1**	7.3	67.6
Danish	37.0	**49.1**	13.9
Dutch	29.1	**46.8**	24.1
English (GB)	22.5	**65.1**	12.4
English (NA)	23.3	**61.8**	15.0
French	28.4	**50.3**	21.4
German	20.8	**58.5**	20.8
Greek	**42.5**	21.4	36.1
Hindi	42.5	36.4	21.1
Hungarian	**56.5**	30.3	13.3
Indonesian	18.9	**57.8**	23.4
Italian	28.1	**49.0**	22.9

Language	LSTM	Hybrid	No pref.
Japanese	**47.4**	28.8	23.9
Korean	**40.6**	25.8	33.5
Mandarin	**48.6**	17.5	33.9
Norwegian	**54.1**	30.8	15.1
Polish	14.6	**75.3**	10.1
Portuguese (BR)	31.4	37.8	30.9
Russian	26.7	**49.1**	24.3
Spanish (ES)	21.0	**47.1**	31.9
Spanish (NA)	22.5	**55.6**	21.9
Swedish	**48.3**	33.6	18.1
Thai	**71.3**	8.8	20.0
Turkish	**61.3**	20.8	18.0
Vietnamese	30.8	30.8	38.5

Acoustic models for speech synthesis – Summary

- **HMM**
  - Discontinuity due to step-wise statistics
  - Difficult to integrate feature extraction
  - Fragmented representation
- **Feedforward NN**
  - Easier to integrate feature extraction
  - Distributed representation
  - Discontinuity due to frame-by-frame independent mapping
- **(LSTM) RNN**
  - Smooth $\rightarrow$ Low latency

Acoustic models for speech synthesis – Future topics

- **Visualization for debugging**
  - Concatenative $\rightarrow$ Easy to debug
  - HMM $\rightarrow$ Hard
  - ANN $\rightarrow$ Harder
- **More flexible voice-based user interface**
  - Concatenative $\rightarrow$ Record all possibilities
  - HMM $\rightarrow$ Weak/rare signals (input) are often ignored
  - ANN $\rightarrow$ Weak/rare signals can contribute
- **Fully integrate feature extraction**
  - Current: Linguistic features $\rightarrow$ Acoustic features
  - Goal: Character sequence $\rightarrow$ Speech waveform

Thanks!

References I

[1] E. Moulines and F. Charpentier.
Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones.
Speech Commn., 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] 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.

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

[5] L. Rabiner.
A tutorial on hidden Markov models and selected applications in speech recognition.
In *Proc. IEEE*, volume 77, pages 257–285, 1989.

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

[7] 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.

[8] 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.

References II

[9] K. Yu, F. Mairesse, and S. Young.
Word-level emphasis modelling in HMM-based speech synthesis.
In *Proc. ICASSP*, pages 4238–4241, 2010.

[10] M. Shannon, H. Zen, and W. Byrne.
Autoregressive models for statistical parametric speech synthesis.
IEEE Trans. Acoust. Speech Lang. Process., 21(3):587–597, 2013.

[11] C. Quillen.
Kalman filter based speech synthesis.
In *Proc. ICASSP*, pages 4618–4621, 2010.

[12] V. Tsiaras, R. Maia, V. Diakoloukas, Y. Stylianou, and V. Digalakis.
Linear dynamical models in speech synthesis.
In *Proc. ICASSP*, pages 300–304, 2014.

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

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

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

[16] K. Nakamura, K. Hashimoto, Y. Nankaku, and K. Tokuda.
Integration of spectral feature extraction and modeling for HMM-based speech synthesis.
IEICE Trans. Inf. Syst., E97-D(6):1438–1448, 2014.

References III

[17] K. Yu, H. Zen, F. Mairesse, and S. Young.
Context adaptive training with factorized decision trees for HMM-based statistical parametric speech synthesis.
Speech Commn., 53(6):914–923, 2011.

[18] H. Zen, M. Gales, Y. Nankaku, and K. Tokuda.
Product of experts for statistical parametric speech synthesis.
IEEE Trans. Audio Speech Lang. Process., 20(3):794–805, 2012.

[19] K. Saino.
A clustering technique for factor analysis-based eigenvoice models.
Master thesis, Nagoya Institute of Technology, 2008.
(in Japanese).

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

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

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

[23] Y. Bengio.
Deep learning: Theoretical motivations.
http://www.iro.umontreal.ca/~bengioy/talks/dlss-3aug2015.pdf, 2015.

References IV

[24] C. Valentini-Botinhao, Z. Wu, and S. King.
Towards minimum perceptual error training for DNN-based speech synthesis.
In *Proc. Interspeech*, pages 869–873, 2015.

[25] S. Takaki, S.-J. Kim, J. Yamagishi, and J.-J. Kim.
Multiple feed-forward deep neural networks for statistical parametric speech synthesis.
In *Interspeech*, pages 2242–2246, 2015.

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

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

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

[29] 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.

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

References V

[31] K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi.
Vector quantization of speech spectral parameters using statistics of dynamic features.
In *Proc. ICSP*, pages 247–252, 1997.

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

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

[34] A. Gutkin, J. Gonzalvo, S. Breuer, and P. Taylor.
Quantized HMMs for low footprint text-to-speech synthesis.
In *Proc. Interspeech*, pages 837–840, 2010.

[35] R. Alvarez, R. Prabhavalkar, and A. Bakhtin.
On the efficient training, representation and execution of deep acoustic models.
In *Proc. Interspeech (submitted)*, 2016.

[36] V. Vanhoucke, M. Devin, and G. Heigold.
Multiframe deep neural networks for acoustic modeling.
In *Proc. ICASSP*, pages 7582–7585. IEEE, 2013.