Acoustic Modeling for Speech Synthesis

Heiga Zen
Dec. 14th, 2015@ASRU
Outline

Background

HMM-based acoustic modeling
  Training & synthesis
  Limitations

ANN-based acoustic modeling
  Feedforward NN
  RNN

Conclusion
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)
Text-to-speech as sequence-to-sequence mapping

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

**Statistical machine translation (SMT)**
Text (discrete symbol sequence) → Text (discrete symbol sequence)
Text-to-speech as sequence-to-sequence mapping

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

**Statistical machine translation (SMT)**
Text (discrete symbol sequence) → Text (discrete symbol sequence)

**Text-to-speech synthesis (TTS)**
Text (discrete symbol sequence) → Speech (real-valued time series)
Speech production process

- Modulation of carrier wave by speech information
  - Frequency transfer characteristics
  - Magnitude start-end
  - Fundamental frequency

Text (concept)

Sound source
- Voiced: pulse
- Unvoiced: noise
This presentation mainly talks about backend
## Concatenative speech synthesis

- Concatenate actual small speech segments from database → *Very high segmental naturalness*
- Single segment per unit (e.g., diphone) → diphone synthesis [1]
- Multiple segments per unit → 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
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?*
Outline

Background

HMM-based acoustic modeling

Training & synthesis
Limitations

ANN-based acoustic modeling

Feedforward NN
RNN

Conclusion
Formulation of SPSS

**Training**
- Extract linguistic features $l$ & acoustic features $o$
- Train acoustic model $\Lambda$ given $(o, l)$

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

Training
- Extract linguistic features $l$ & acoustic features $o$
- Train acoustic model $\Lambda$ given $(o, l)$

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

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

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

Training
- Extract linguistic features $l$ & acoustic features $o$
- Train acoustic model $\Lambda$ given $(o, l)$

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

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

$$\hat{o} = \arg \max_o p(o \mid l, \hat{\Lambda})$$
### Training – HMM-based acoustic modeling

$$p(o \mid l, \Lambda) = \sum_{\forall q} p(o \mid q, \Lambda)P(q \mid l, \Lambda) \quad q: \text{hidden states}$$

$$= \sum_{\forall q} \prod_{t=1}^{T} p(o_t \mid q_t, \Lambda)P(q \mid l, \Lambda) \quad q_t: \text{hidden state at } t$$

$$= \sum_{\forall q} \prod_{t=1}^{T} \mathcal{N}(o_t; \mu_{q_t}, \Sigma_{q_t})P(q \mid l, \Lambda)$$

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

→ Impossible to have enough data to cover all combinations

```
L=voice?
 yes
 no
yes yes
no
no no

R=silence?
yes
no

L=“gy”?  
yes
no

Leaf nodes

Synthesized Gaussians
```

```
k-a+b/A=1/... 
...

t-e+n/A=0/... 
...

t-e+n/A=0/... 
...

w-a+sil/A=0/... 
...

w-a+t/A=0/... 
...

gy-e+sil/A=0/... 
...

gy-a+pau/A=0/... 
...

g-e+sil/A=1/...
```
Training – Example

Acoustic features $o$

Mean sequence $\mu$

$q$

sil  j  i  b  u  N  n  o  j  i  t  s  u  r  y  o  k  u  w  a  sil
Formulation of SPSS

Training
- Extract linguistic features $l$ & acoustic features $o$
- Train acoustic model $\Lambda$ given $(o, l)$

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

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

$$\hat{o} = \arg \max_{o} p(o | l, \hat{\Lambda})$$
Synthesis – Predict most probable acoustic features

\[ \hat{o} = \arg\max_o p(o \mid l, \hat{\Lambda}) \]

\[ = \arg\max_o \sum_{\forall q} p(o, q \mid l, \hat{\Lambda}) \]

\[ \approx \arg\max_o \max_q p(o, q \mid l, \hat{\Lambda}) \]

\[ = \arg\max_o \max_q p(o \mid q, \hat{\Lambda}) P(q \mid l, \hat{\Lambda}) \]

\[ \approx \arg\max_o p(o \mid \hat{q}, \hat{\Lambda}) \quad \text{s.t.} \quad \hat{q} = \arg\max_q P(q \mid l, \hat{\Lambda}) \]

\[ = \arg\max_o \mathcal{N}(o; \mu_{\hat{q}}, \Sigma_{\hat{q}}) \]

\[ = \mu_{\hat{q}} \]

\[ = \left[ \mu_{\hat{q}_1}, \ldots, \mu_{\hat{q}_T} \right]^\top \]
\( \hat{o} \rightarrow \text{step-wise} \rightarrow \text{discontinuity can be perceived} \)

\[
O_t = \begin{bmatrix} c_t^T, \Delta c_t^T \end{bmatrix}^T
\]

\[
\Delta c_t = c_t - c_{t-1}
\]

\[
O_{t-1} = \begin{bmatrix} c_{t-1}, \Delta c_{t-1} \end{bmatrix}
\]

\[
O_t = \begin{bmatrix} c_t, \Delta c_t \end{bmatrix}
\]

\[
O_{t+1} = \begin{bmatrix} c_{t+1}, \Delta c_{t+1} \end{bmatrix}
\]

\[
W = \begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & 0 & I & 0 & 0 \\
\cdots & -I & I & 0 & 0 \\
\cdots & 0 & 0 & I & 0 \\
\cdots & 0 & -I & I & 0 \\
\cdots & 0 & 0 & 0 & I \\
\cdots & 0 & 0 & -I & I \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix}
\]

\[
c = \begin{bmatrix} c_{t-2} \\
\cdots \\
c_{t-1} \\
c_t \\
c_{t+1} \\
c_{t+2} \\
\cdots \\
\end{bmatrix}
\]
\[ \hat{o} = \arg \max_{o} p(o \mid \hat{q}, \hat{\Lambda}) \quad s.t. \quad o = Wc \]

\[ \hat{c} = \arg \max_{c} \mathcal{N}(Wc; \mu_{\hat{q}}, \Sigma_{\hat{q}}) \]

\[ = \arg \max_{c} \log \mathcal{N}(Wc; \mu_{\hat{q}}, \Sigma_{\hat{q}}) \]
Synthesis – Speech parameter generation algorithm [7]

\[ \hat{o} = \arg \max_o p(o \mid \hat{q}, \hat{\Lambda}) \quad s.t. \quad o = Wc \]

\[ \hat{c} = \arg \max_c \mathcal{N}(Wc; \mu\hat{q}, \Sigma\hat{q}) \]

\[ = \arg \max_c \log \mathcal{N}(Wc; \mu\hat{q}, \Sigma\hat{q}) \]

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

\[ W^\top \Sigma^{-1}_{\hat{q}} Wc = W^\top \Sigma^{-1}_{\hat{q}} \mu\hat{q} \]

where

\[ \mu_q = [\mu_{q1}^\top, \mu_{q2}^\top, \ldots, \mu_{qT}^\top]^\top \]

\[ \Sigma_q = \text{diag} [\Sigma_{q1}, \Sigma_{q2}, \ldots, \Sigma_{qT}] \]
Synthesis – Speech parameter generation algorithm [7]

\[
W^T \Sigma^{-1} \hat{q} W^T = W^T \Sigma^{-1} \hat{q} \mu \hat{q}
\]

Heiga Zen
Acoustic Modeling for Speech Synthesis
Dec. 14th, 2015  21 of 62
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
  → Step-wise statistics
- Using dynamic feature constraints
  → 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 → Inefficient to represent dependency betw. ling. & acoust. feats.
- Minor features are never used (e.g., word-level emphasis [9]) → Little or no effect
Alternatives – Stepwise statistics

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

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 → Data fragmentation
Often employ trees from HMM → 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
Outline

Background

HMM-based acoustic modeling
  Training & synthesis
  Limitations

ANN-based acoustic modeling
  Feedforward NN
  RNN

Conclusion
Linguistic → 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 $\rightarrow$ 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
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

\[
\hat{o}_t = \arg \min_\Lambda \sum_t \|o_t - \hat{o}_t\|_2 \\
\Lambda = \{W_{hl}, W_{oh}, b_h, b_o\}
\]

\[
\hat{o}_t \approx \mathbb{E}[o_t | l_t] \rightarrow \text{Replace decision trees & Gaussian distributions}
\]

Frame-level linguistic feature \(l_t\)
Frame-level acoustic feature \(o_t\)

Target

Input

\[
h_t = f(W_{hl}l_t + b_h) \\
\hat{o}_t = W_{oh}h_t + b_o
\]
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
  → 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

- **Input layer**
  - Duration prediction
  - Input feature extraction
  - Text analysis

- **Hidden layers**
  - Input features including binary & numeric features at frame $t$

- **Output layer**
  - Statistics (mean & var) of speech parameter vector sequence

- **SPEECH**
  - Waveform synthesis
  - Parameter generation

- **Input features**
  - Binary features
  - Numeric features
  - Duration feature
  - Frame position feature

- **Spectral features**
  - Excitation features
  - V/UV feature

Heiga Zen
Acoustic Modeling for Speech Synthesis
Dec. 14th, 2015
DNN-based speech synthesis [21] – Example

![Graph showing 5-th Mel-cepstrum over Frames]

- **Natural speech**
- **DNN (smoothed)**

Heiga Zen
Acoustic Modeling for Speech Synthesis
Dec. 14th, 2015

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

<table>
<thead>
<tr>
<th>Preference scores (higher one is better)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>15.8%</td>
</tr>
<tr>
<td>16.1%</td>
</tr>
<tr>
<td>12.7%</td>
</tr>
</tbody>
</table>
Feedforward NN-based acoustic model – Limitation

Each frame is mapped independently → Smoothing is still essential

<table>
<thead>
<tr>
<th>Preference scores (higher one is better)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN with dyn</td>
</tr>
<tr>
<td>67.8%</td>
</tr>
</tbody>
</table>
Feedforward NN-based acoustic model – Limitation

Frame-level linguistic feature $l_t$
Frame-level acoustic feature $o_t$
Input

Target

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

Preference scores (higher one is better)

<table>
<thead>
<tr>
<th></th>
<th>DNN with dyn</th>
<th>DNN without dyn</th>
<th>No pref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result</td>
<td>67.8%</td>
<td>12.0%</td>
<td>20.0%</td>
</tr>
</tbody>
</table>

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

Heiga Zen
Acoustic Modeling for Speech Synthesis
Dec. 14th, 2015
RNN-based acoustic model [28, 29]

\[ h_t = f(W_{hl}l_t + W_{hh}h_{t-1} + b_h) \]
\[ \hat{o}_t = W_{oh}h_t + b_o \]
\[ \hat{\Lambda} = \arg \min_{\Lambda} \sum_t \|o_t - \hat{o}_t\|_2 \quad \Lambda = \{W_{hl}, W_{hh}, W_{oh}, b_h, b_o\} \]

- **DNN:** \[ \hat{o}_t \approx E[ o_t | l_t ] \]
- **RNN:** \[ \hat{o}_t \approx E[ o_t | l_1, \ldots, l_t ] \]
RNN-based acoustic model [28, 29]

- Only able to use previous contexts
  → Bidirectional RNN [27]: $\hat{o}_t \approx \mathbb{E} [o_t \mid l_1, \ldots, l_T]$
RNN-based acoustic model [28, 29]

- Only able to use previous contexts
  → Bidirectional RNN [27]: \( \hat{o}_t \approx \mathbb{E} [o_t | l_1, \ldots, l_T] \)

- Trouble accessing long-range contexts
  - Information in hidden layers loops quickly decays over time
  - Prone to being overwritten by new information from inputs
  → 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

<table>
<thead>
<tr>
<th></th>
<th>DNN with dyn</th>
<th>LSTM with dyn</th>
<th>No pref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN with dyn</td>
<td>18.4%</td>
<td>34.9%</td>
<td>47.6%</td>
</tr>
<tr>
<td>LSTM with dyn</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No pref.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Smoothing was still effective
LSTM-RNN-based acoustic model [29]
Subjective preference test (same US English data)

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

<table>
<thead>
<tr>
<th></th>
<th>DNN with dyn</th>
<th>LSTM with dyn</th>
<th>No pref.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18.4%</td>
<td>34.9%</td>
<td>47.6%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>LSTM with dyn</th>
<th>LSTM without dyn</th>
<th>No pref.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>21.0%</td>
<td>12.2%</td>
<td>66.8%</td>
</tr>
</tbody>
</table>

→ Smoothing was still effective
Why?

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

Gate output: 0 → 1

Input gate == 1
→ Write memory

Forget gate == 0
→ Reset memory

Output gate == 1
→ Read memory

Heiga Zen
Acoustic Modeling for Speech Synthesis
Dec. 14th, 2015
How?

- Using loss function incorporating continuity
How?

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

\[ h_t = \text{LSTM}(l_t) \quad \hat{o}_t = W_{oh}h_t + W_{oo}\hat{o}_{t-1} + b_o \]
How?

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

\[ h_t = \text{LSTM} \left( l_t \right) \quad \hat{o}_t = W_{oh}h_t + W_{oo}\hat{o}_{t-1} + b_o \]

Works pretty well

<table>
<thead>
<tr>
<th></th>
<th>LSTM with dyn (Feedforward)</th>
<th>LSTM without dyn (Recurrent)</th>
<th>No pref.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>21.8%</td>
<td>21.0%</td>
<td>57.2%</td>
</tr>
</tbody>
</table>
How?

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

\[ h_t = \text{LSTM}(l_t) \quad \hat{o}_t = W_{oh}h_t + W_{oo}\hat{o}_{t-1} + b_o \]

Works pretty well

<table>
<thead>
<tr>
<th></th>
<th>LSTM with dyn (Feedforward)</th>
<th>LSTM without dyn (Recurrent)</th>
<th>No pref.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>21.8%</td>
<td>21.0%</td>
<td>57.2%</td>
</tr>
</tbody>
</table>

Having two smoothing together doesn’t work well → Oversmoothing?

<table>
<thead>
<tr>
<th></th>
<th>LSTM with dyn (Recurrent)</th>
<th>LSTM without dyn (Recurrent)</th>
<th>No pref.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16.6%</td>
<td>29.2%</td>
<td>54.2%</td>
</tr>
</tbody>
</table>
Low-latency TTS by unidirectional LSTM-RNN [29]

HMM / DNN

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

$$W^\top \Sigma_{\hat{q}}^{-1} W c = W^\top \Sigma_{\hat{q}}^{-1} \mu_{\hat{q}} \quad T: \text{Utterance length}$$
Low-latency TTS by unidirectional LSTM-RNN [29]

HMM / DNN

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

$$W^\top \Sigma_{\hat{q}}^{-1} W c = W^\top \Sigma_{\hat{q}}^{-1} \mu_{\hat{q}} \quad T: \text{Utterance length}$$

- Order of operations to determine the first frame $c_1$ (latency)
  - Cholesky decomposition [7] $\rightarrow \mathcal{O}(T)$
  - Recursive approximation [31] $\rightarrow \mathcal{O}(L) \quad L: \text{lookahead, 10 \sim 30}$
Low-latency TTS by unidirectional LSTM-RNN [29]

HMM / DNN

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

$$W^\top \Sigma_{\hat{q}}^{-1} W c = W^\top \Sigma_{\hat{q}}^{-1} \mu_{\hat{q}} \quad T: \text{Utterance length}$$

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

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

Acoustic feature prediction LSTM

Duration prediction LSTM

<table>
<thead>
<tr>
<th>phoneme</th>
<th>h</th>
<th>e</th>
<th>l</th>
<th>ou</th>
</tr>
</thead>
<tbody>
<tr>
<td>syllable</td>
<td>h e2</td>
<td>l ou1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>word</td>
<td>hello</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Low-latency TTS by LSTM-RNN [29] – Implementation

Acoustic feature prediction LSTM

Duration prediction LSTM

Linguistic features (phoneme)

Feature functions

phoneme

syllable

word

<table>
<thead>
<tr>
<th>phoneme</th>
<th>h</th>
<th>e</th>
<th>l</th>
<th>ou</th>
</tr>
</thead>
<tbody>
<tr>
<td>syllable</td>
<td>h e2</td>
<td>l ou1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>word</td>
<td>hello</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Low-latency TTS by LSTM-RNN [29] – Implementation

Acoustic feature prediction LSTM

Duration prediction LSTM

Linguistic features (phoneme)

Feature functions

Durations (targets)

<table>
<thead>
<tr>
<th>phoneme</th>
<th>h</th>
<th>e</th>
<th>l</th>
<th>ou</th>
</tr>
</thead>
<tbody>
<tr>
<td>syllable</td>
<td>h e2</td>
<td></td>
<td></td>
<td>l ou1</td>
</tr>
<tr>
<td>word</td>
<td></td>
<td></td>
<td>hello</td>
<td></td>
</tr>
</tbody>
</table>
Low-latency TTS by LSTM-RNN [29] – Implementation

Acoustic feature prediction LSTM

Linguistic features (frame)

Durations (targets) 9

Duration prediction LSTM

Linguistic features (phoneme)

Feature functions

<table>
<thead>
<tr>
<th>phoneme</th>
<th>h</th>
<th>e</th>
<th>l</th>
<th>ou</th>
</tr>
</thead>
<tbody>
<tr>
<td>syllable</td>
<td>h e2</td>
<td>l ou1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>word</td>
<td>hello</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Low-latency TTS by LSTM-RNN [29] – Implementation

Acoustic feature prediction LSTM

Linguistic features (frame)

Durations (targets)

9

Duration prediction LSTM

Linguistic features (phoneme)

Feature functions

phoneme

syllable

word

Acoustic features (targets)

h e l ou
h e2 l ou1
hello
phoneme
syllable
word

Heiga Zen
Acoustic Modeling for Speech Synthesis
Low-latency TTS by LSTM-RNN [29] – Implementation

- Waveform
- Acoustic features (targets)
- Acoustic feature prediction LSTM
- Linguistic features (frame)
- Durations (targets)
- Duration prediction LSTM
- Linguistic features (phoneme)
- Feature functions
- Phoneme:
  - h
  - e
  - l
  - ou
- Syllable:
  - h e2
  - l ou1
- Word:
  - hello
Low-latency TTS by LSTM-RNN [29] – Implementation

Waveform

Acoustic features (targets)

Acoustic feature prediction LSTM

Linguistic features (frame)

Durations (targets) 9

Duration prediction LSTM

Linguistic features (phoneme)

Feature functions

phoneme

syllable

word

<table>
<thead>
<tr>
<th></th>
<th>h</th>
<th>e</th>
<th>l</th>
<th>ou</th>
</tr>
</thead>
<tbody>
<tr>
<td>phoneme</td>
<td>h e2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>syllable</td>
<td></td>
<td></td>
<td>l ou1</td>
<td></td>
</tr>
<tr>
<td>word</td>
<td></td>
<td>hello</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Low-latency TTS by LSTM-RNN [29] – Implementation

Acoustic feature prediction LSTM

Duration prediction LSTM

Linguistic features (phoneme) ⇒ Feature functions

Durations (targets) 9

Linguistic features (frame)

Acoustic features (targets)

Waveform

phoneme

syllable

word

hello

h e2

l ou1

h e l ou

l ou

Heiga Zen

Acoustic Modeling for Speech Synthesis
Low-latency TTS by LSTM-RNN [29] – Implementation

Waveform
Acoustic features (targets)

Acoustic feature prediction LSTM
Linguistic features (frame)
Durations (targets)
Duration prediction LSTM
Linguistic features (phoneme)

Feature functions
phoneme
syllable
word

||| ||
---|---|---|---|
phoneme | h | e | l | ou
syllable | h e2 | l ou1
word | hello
Low-latency TTS by LSTM-RNN [29] – Implementation

Waveform

Acoustic features (targets)

Acoustic feature prediction LSTM

Linguistic features (frame)

Durations (targets)

Duration prediction LSTM

Linguistic features (phoneme)

Feature functions

phoneme

syllable

word

h e l ou

h e2 l ou1

hello

phoneme

syllable

word

Linguistic Structure

Acoustic feature prediction LSTM

Durations (targets)

12

9

Linguistic features (frame)

Acoustic features (targets)

Waveform

⇒

Feature functions

phoneme

syllable

word

h e l ou

h e2 l ou1

hello

Heiga Zen

Acoustic Modeling for Speech Synthesis

Dec. 14th, 2015
Low-latency TTS by LSTM-RNN [29] – Implementation

Waveform

Acoustic features (targets)

Acoustic feature prediction LSTM

Linguistic features (frame)

Durations (targets) 9 12 10

Duration prediction LSTM

Linguistic features (phoneme)

Feature functions

phoneme

syllable

word

h e l ou

h e2 l ou1

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

Waveform

Acoustic features (targets)

Acoustic feature prediction LSTM

Linguistic features (frame)

Durations (targets) 9 12 10 10

Duration prediction LSTM

Linguistic features (phoneme)

Feature functions

phoneme h e l ou

syllable h e2 l ou1

word hello

Heiga Zen

Acoustic Modeling for Speech Synthesis

Dec. 14th, 2015 46 of 62
Some comments

Is this new? … no

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

Google Text-to-speech

This app is compatible with all of your devices.

Heiga Zen
Acoustic Modeling for Speech Synthesis
Dec. 14th, 2015
Network architecture

- ~ 400 sparse input
- FF / ReLU
- LSTMP
- LSTMP
- LSTMP
- RNN / Linear

⇐ Embed to continuous space
⇐ Encourage smooth trajectory
Results – HMM / LSTM-RNN

Subjective 5-scale Mean Opinion Score test (i18n)

Better MOS

<table>
<thead>
<tr>
<th>Language</th>
<th>HMM</th>
<th>LSTM-RNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmn-CN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>da-DK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>de-DE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>en-IN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>en-US</td>
<td></td>
<td></td>
</tr>
<tr>
<td>es-ES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>es-US</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fr-FR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hi-IN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id-ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ja-JP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ko-KR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nl-NL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pt-BR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ru-RU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>th-TH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tr-TR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>yue-HK</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results – HMM / LSTM-RNN

Subjective preference test (i18n)
# Results – HMM / LSTM-RNN

## Latency & Battery/CPU usage

### Latency (Nexus 7 2013)

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Average/Max latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>very short (1 character)</td>
<td>26/30</td>
</tr>
<tr>
<td>short (~30 characters)</td>
<td>123/172</td>
</tr>
<tr>
<td>long (~80 characters)</td>
<td>311/418</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>HMM</td>
<td>37/72</td>
</tr>
<tr>
<td>LSTM-RNN</td>
<td>63/88</td>
</tr>
</tbody>
</table>

### CPU usage
- HMM $\rightarrow$ LSTM-RNN: +48%

### Battery usage (Daily usage by a blind Googler)
- HMM: 2.8% of 1475 mAH $\rightarrow$ LSTM-RNN: 4.8% of 1919 mAH
Results – HMM / LSTM-RNN

Summary

- **Naturalness**
  - LSTM-RNN > HMM

- **Latency**
  - LSTM-RNN < HMM

- **CPU/Battery usage**
  - LSTM-RNN > HMM

LSTM-RNN-based TTS is in production at Google
Outline

Background

HMM-based acoustic modeling
  Training & synthesis
  Limitations

ANN-based acoustic modeling
  Feedforward NN
  RNN

Conclusion
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

(LSTM) RNN
- Smooth → Low latency
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
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 → Easy to debug
  - HMM → Hard
  - ANN → Harder

- More flexible voice-based user interface
  - Concatenative → Record all possibilities
  - HMM → Weak/rare signals (input) are often ignored
  - ANN → Weak/rare signals can contribute

- Fully integrate feature extraction
  - Current: Linguistic features
    → Acoustic features
  - Goal: Character sequence
    → Speech waveform
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
• **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
References I

Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones.

Unit selection in a concatenative speech synthesis system using a large speech database.

Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis.

Statistical parametric speech synthesis.

A tutorial on hidden Markov models and selected applications in speech recognition.

The use of context in large vocabulary speech recognition.

Speech parameter generation algorithms for HMM-based speech synthesis.

Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic features.
Word-level emphasis modelling in HMM-based speech synthesis.

Autoregressive models for statistical parametric speech synthesis.

Kalman filter based speech synthesis.

Linear dynamical models in speech synthesis.

Statistical approach to vocal tract transfer function estimation based on factor analyzed trajectory hmm.

Minimum generation error training with direct log spectral distortion on LSPs for HMM-based speech synthesis.

Statistical parametric speech synthesis with joint estimation of acoustic and excitation model parameters.

Integration of spectral feature extraction and modeling for HMM-based speech synthesis.
Context adaptive training with factorized decision trees for HMM-based statistical parametric speech synthesis.  

Product of experts for statistical parametric speech synthesis.  

*A clustering technique for factor analysis-based eigenvoice models.*  

Statistical parametric speech synthesis based on speaker and language factorization.  


Distributed representation.  

[23] Y. Bengio.  
Deep learning: Theoretical motivations.  


