Improving LPCNet-based Text-to-Speech with Linear Prediction-structured Mixture Density Network

Min-Jae Hwang\textsuperscript{1,2}, Eunwoo Song\textsuperscript{3}, Ryuichi Yamamoto\textsuperscript{4}, Frank Soong\textsuperscript{5} and Hong-Goo Kang\textsuperscript{2}

\textsuperscript{1}Search Solutions Inc., South Korea
\textsuperscript{2}Yonsei Univ., South Korea, \textsuperscript{3}NAVER Corp., South Korea
\textsuperscript{4}Line Corp., Japan, \textsuperscript{5}Microsoft Research Asia, China
OVERVIEW

Paper objective

• Improving the quality of LPCNet-based parametric speech synthesis system

Proposed systems

• LP-MDN: Linear prediction-structured mixture density network
  • Structurally merge the LP process with an autoregressive neural vocoding framework
• iLPCNet: Improved LPCNet vocoder
  • Incorporating LP-MDN into LPCNet framework
• Effective training and generation methods

Performance

[MOS test result] [Preference test result]
CONTENTS

Introduction
  • LPCNet-based neural vocoding [1]

Proposed system
  • Linear prediction-structured mixture density network
  • Improved LPCNet vocoder
  • Effective training and generation methods

Experiments
  • Performance evaluations

Summary & Conclusion

**LPCNet-based Neural Vocoding**

Incorporate linear prediction (LP) structure within WaveRNN framework

---

**Characteristics**

- WaveRNN architecture
  - Accelerate the generation speed of autoregressive neural vocoder
- LP synthesis-based spectral shaping filter
  - Achieve good synthesis quality by attenuating quantization noise caused by $\mu$-law modeling
- Various tuning methods for $\mu$-law modeling
  - Waveform embedding, discrete training noise injection, conditional sampling for softmax distribution, pre-emphasis filter, ...

---

[LP synthesis process]

\[
p_n = \sum_{i=1}^{p} a_i x_{n-i},
\]

\[
x_n = e_n + p_n
\]
**LPCNet-based Neural Vocoder**

Incorporate linear prediction (LP) structure within WaveRNN framework

Methods to improve performance

- Replace the $\mu$-law waveform model with a continuous waveform model
  - Improve synthesis quality by utilizing densely distributed waveform sample
  - Simplify the tuning methods

[LP synthesis process]

\[ p_n = \sum_{i=1}^{p} \alpha_i x_{n-i}, \]

\[ x_n = e_n + p_n \]
**LPCNet-based Neural Vocoding**

Incorporate linear prediction (LP) structure within WaveRNN framework

**Methods to improve performance**

- Replace the \( \mu \)-law waveform model with a continuous waveform model
  - Improve synthesis quality by utilizing densely distributed waveform sample
  - Simplify the tuning methods
- Suggest a closed-loop solution of LP structure for compact representation

\[
P_n = \sum_{i=1}^{p} \alpha_i x_{n-i},
\]

\[
x_n = e_n + p_n
\]
**LP-STRUCTURED MDN**

**Basic assumption on autoregressive neural vocoder**

1. Previous speech samples, \( x_{<n} \), are given
2. LP coefficients, \( \{\alpha_{n,i}\} \), are given

Their linear combination, \( p_n = \sum_{i=1}^{P} \alpha_{n,i} x_{n-i} \), are also given

**Probabilistic analysis**

\[
x_n = e_n + p_n
\]

\[
X_n | (x_{<n}, h) = E_n | (x_{<n}, h) + p_n
\]

Random variables \( X_n \) and \( E_n \) are depends on only the constant difference of \( p_n \)

**Mixture of Gaussian (MoG) modeling**

\[
p(x_n | x_{<n}, h_n) = \sum_{n=1}^{N} \omega_n \cdot \frac{1}{\sqrt{2\pi s_{n,d}}} \exp \left[ \frac{(x_n - \mu_{n,i})^2}{2s_{n,i}^2} \right]
\]

- Utilize the shifting property of \(^2\) order random variable
  \[
  \omega_i^* = \omega_i^c \\
  \mu_i^* = \mu_i^c + p_n \\
  s_i^* = s_i^c
  \]

Difference between speech and excitation's mixture parameters are only mean parameters
LP-STRUCTURED MDN

LP-MDN-based neural vocoding

1. Mixture parameter prediction
   \[
   [z_n^m, z_n^p, z_n^s] = \text{NeuralVocoder}(x_{<n}, h_n)
   \]

2. Compute prediction term
   \[p_n = \sum_{i=1}^{p} \alpha_{n,i} x_{n-i}\]

3. Mixture parameter modification
   \[
   \omega_n = \text{softmax}(z_n^m), \\
   \mu_n = z_n^\mu + p_n, \\
   s_n = \exp(z_n^s)
   \]

4. MoG likelihood calculation
   \[
   p(x_n | x_{<n}, h_n) = \sum_{i=1}^{n} \omega_{n,i} \cdot \frac{1}{\sqrt{2\pi} s_{n,i}} \exp\left[-\frac{(x_n - \mu_{n,i})^2}{2s_{n,i}^2}\right]
   \]

5. Train the network to minimize negative log-likelihood loss
   \[
   L_{nll} = \sum_n [-\log p(x_n | x_{<n}, h_n)]
   \]
Improved LPCNet

**iLPCNet Vocoder**

**Upsampling network**

- Match the time-resolution of acoustic features to the sampling rate of speech signal
  
  - **Architecture**
    - Two stacks of convolution layer
      - Extract contextual information of acoustic features
    - Transposed convolution layer
      - Upsample the context features

**Waveform generation network**

- Autoregressively generate waveform samples

  - **Architecture**
    - Two stacks of gated recurrent unit (GRU) layers
    - Apply LP-MDN to generate the speech's distribution
**Effective Training and Generation Methods**

**Short-time Fourier transform (STFT)-based power loss**

\[ L_{pl} = \left\| STFT(x) - STFT(\hat{x}) \right\|^2_2 \]

- \( L = L_{all} + \lambda L_{pl} \)

- Capture the time-frequency distribution of the speech waveform

**Continuous training noise injection**

\[ \hat{x}_{n+1} = x_{n+1} + \frac{4}{2^{16}} \varepsilon, \text{ where } \varepsilon \sim N(0,1) \]

- \( x_n = iLPCNet(\hat{x}_{n-1}, h_n) \)

- Train the propagated prediction error via autoregressive connection

- Simplify complicated noise injection pipeline of original LPCNet

[Noise injection process of **LPCNet**]

[Noise injection process of **iLPCNet**]
Effective Training and Generation Methods

Conditional sampling for MoG distribution

- Conventional random sampling method
  \[ x_{\text{rand}} \sim N(\mu, s) \]
  - Noisy artifacts in the voiced region

- Distribution sharpening method
  \[ x_{\text{sharp}} \sim N(\mu, c \cdot s), \text{ where } c < 1 \]
  - Eliminate noisy artifacts by reducing noise component

- Proposed conditional sampling method
  \[ x = \text{vuv} \cdot x_{\text{sharp}} + (1 - \text{vuv}) \cdot x_{\text{rand}} \]
  - Sharpened sampling at the voiced region
  - Random sampling at the unvoiced region

[Spectrogram example of conditional sampling]
## Comparison with Original LPCNet

<table>
<thead>
<tr>
<th></th>
<th>LPCNet</th>
<th>Proposed iLPCNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Distribution type</strong></td>
<td>Discrete</td>
<td>Continuous</td>
</tr>
<tr>
<td><strong>Method to reflect LP structure</strong></td>
<td>Feeding LP-related signals, ([e_{n-1}, x_{n-1}, p_n]), into GRU</td>
<td>LP-MDN</td>
</tr>
<tr>
<td></td>
<td>Open-loop solution</td>
<td>Closed-loop solution</td>
</tr>
<tr>
<td><strong>Target of WaveRNN</strong></td>
<td>Excitation</td>
<td>Speech</td>
</tr>
<tr>
<td><strong>Tuning methods</strong></td>
<td>Waveform embedding</td>
<td>STFT-based power loss</td>
</tr>
<tr>
<td></td>
<td><strong>Discrete</strong> noise injection</td>
<td><strong>Continuous</strong> noise injection</td>
</tr>
<tr>
<td></td>
<td>Conditional sharpening for <strong>softmax</strong> distribution</td>
<td>Conditional sharpening for <strong>MoG</strong> distribution</td>
</tr>
</tbody>
</table>
**EXPERIMENT SETUP**

Common settings

<table>
<thead>
<tr>
<th>Database</th>
<th>Korean professional female</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling rate / Quantization bit</td>
<td>24kHz / 16 bits</td>
</tr>
<tr>
<td>Training / validation / test</td>
<td>4,976 (9.9 hours) / 280 / 140</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Acoustic features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extracted by ITFTE vocoder [1]</td>
</tr>
<tr>
<td>79-dim.</td>
</tr>
<tr>
<td>5-ms (=120 samples) frame shift</td>
</tr>
<tr>
<td>Zero mean &amp; unit variance normalization</td>
</tr>
</tbody>
</table>

Neural vocoders

- WaveNet [2]
- LPCNet [3]
- Proposed iLPCNet

Scenarios

- Analysis / synthesis (A/S) scenario
- Text-to-speech (TTS) scenario
  - Tacotron 2 acoustic model [4]

Performance evaluation

- Mean opinion score (MOS) listening test
- A-B preference test

---

## Experiment Setup

### Neural vocoders

- **WaveNet vocoder**

  | Dilation | 3 * [1, 2, 4, 8, 16, 32, 64, 128, 256, 512] |
  | Layer    | 30 |
  | Receptive field | 3,071 |
  | Skip channels    | 128 |
  | Residual channels | 128 |

- **LPCNet vocoder**

  | FC layer dimension | 64 |
  | GRU A dimension    | 256 |
  | GRU B dimension    | 16 |
  | Waveform embedding dimension | 256 |

- **Proposed iLPCNet vocoder**

  | FC layer dimension | 256 |
  | Transposed convolution kernel size | 120 (5-ms) |
  | GRU A dimension    | 256 |
  | GRU B dimension    | 16 |
  | Speech distribution | Single Gaussian distribution |
  | Power loss weight, $\lambda$ | 10.0 |
  | Sharpening factor, $c$ | 0.7 |

- Same GRU size with LPCNet vocoder
## Experiment Setup

**Tacotron 2 acoustic model for TTS scenario**

<table>
<thead>
<tr>
<th>Encoder</th>
<th>Character embedding</th>
<th>Dimension</th>
<th>512</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution layer</td>
<td>Number of layers</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kernel size</td>
<td>10×1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Channels</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>BiLSTM layer</td>
<td>Units</td>
<td>512</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attention</th>
<th>Location-sensitive attention</th>
<th>Dimension</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Kernel size</td>
<td>64×1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decoder</th>
<th>Pre-net FC layer</th>
<th>Number of layers</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dimension</td>
<td>256</td>
<td></td>
</tr>
<tr>
<td>LSTM layer</td>
<td>Number of layers</td>
<td>1,024</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Units</td>
<td>1,024</td>
<td></td>
</tr>
<tr>
<td>Post-net convolution layer</td>
<td>Number of layers</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kernel size</td>
<td>5×1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Channels</td>
<td>512</td>
<td></td>
</tr>
</tbody>
</table>
PERFORMANCE EVALUATIONS

MOS test

- Score the quality of speech
- 15 native Korean listeners
- 15 randomly selected synthesized utterances from test set

Results

<table>
<thead>
<tr>
<th>Score</th>
<th>Quality</th>
<th>Impairment</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Excellent</td>
<td>Imperceptible</td>
</tr>
<tr>
<td>4</td>
<td>Good</td>
<td>Perceptible but not annoying</td>
</tr>
<tr>
<td>3</td>
<td>Fair</td>
<td>Slightly annoying</td>
</tr>
<tr>
<td>2</td>
<td>Poor</td>
<td>Annoying</td>
</tr>
<tr>
<td>1</td>
<td>Bad</td>
<td>Very annoying</td>
</tr>
</tbody>
</table>

[Scoring criteria for MOS test]
PERFORMANCE EVALUATIONS

A-B preference test

- Rate the quality preference
- 15 native Korean listeners
- 15 randomly selected synthesized utterances from test set

Results

<table>
<thead>
<tr>
<th></th>
<th>LPCNet</th>
<th>iLPCNet (ours)</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/S</td>
<td>33.3 %</td>
<td>42.5 %</td>
<td>24.2 %</td>
</tr>
<tr>
<td>TTS</td>
<td>20.8 %</td>
<td>48.3 %</td>
<td>30.8 %</td>
</tr>
</tbody>
</table>

p-value = 0.06

p-value < 10^{-10}
**SUMMARY & CONCLUSION**

**Summary**
- Proposed an improved LPCNet (iLPCNet) vocoder-based parametric TTS system

**Linear prediction (LP)-structured mixture density network (MDN)**
- Structurally constructed the LP structure within an autoregressive neural vocoder framework

**Improved LPCNet vocoder**
- Incorporated LP-MDN into LPCNet vocoder with additional effective training and generation methods
- Achieved simpler and more compact architecture by removing extra modules in LPCNet, which was designed for handling the quantization effect caused by $\mu$-law method

**Performance evaluation results**
- Outperformed the conventional neural vocoding systems
  - 4.41 MOS result
  - 27.5% higher quality preference than conventional LPCNet vocoder
Thank you!