Noise Data Augmentation For Speaker Recognition Using Conditional Generative Adversarial Networks

Peiyao Sheng, Yanmin Qian
Shanghai Jiao Tong University

Dec. 14, 2017
SPEECH AND SECURITY

Automatic Speech Recognition
Text to Speech (TTS)
Speaker Recognition, SR

......

privacy protection
identification
verification
restricted services

......
SPEAKER RECOGNITION

- Verification
  - Yes or No
  - 1-1
- Identification
  - Which Is Correct
  - n-1
- text-dependent (TD-SV)
  - Fixed password system
  - ...
- text-independent (TI-SV)
  - Long-term statistic based system
  - ...
- Enrollment —> Recognition
  - Clean Data —> Noisy Environment
  - More Data!
Generative Adversarial Networks

- NOISE
- G
- FAKE IMAGE
- REAL IMAGE
- REAL?
- REAL?
- D

- $D(x)$ tries to be near 1
- $D(G(z))$ near 0, $G$ tries to make $D(G(z))$ near 1
- Differentiable function $D$
- $x$ sampled from data
- $D$
- $z$ sampled from model
- Differentiable function $G$
- Input noise $z$
Generative Adversarial Networks

- Generator $G$
  - $G : z \mapsto x$
  - $G(z; \theta^{(G)})$
  - $J^{(G)}(\theta^{(D)}, \theta^{(G)})$
  - $p_g(x)$ hard to compute

- Discriminator $D$
  - $D : x \mapsto y$
  - $D(x; \theta^{(D)})$
  - $J^{(D)}(\theta^{(D)}, \theta^{(G)})$
  - $J^{(D)}$ evaluate difference

Two-player minimax game
- Value function $V(G, D)$
- $G^* = \text{arg min}_G \text{max}_D V(D, G)$

$$V = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x \sim p_g(x)}[\log(1 - D(x))]$$
Conditional Generative Adversarial Networks

\[
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)} [\log (1 - D(G(z|y)))].
\]
Figure 1: Generator (G) and discriminator (D) in the framework. G generates an noisy speech from a clean input to fool D. D tries to discriminate an input as real or generated, conditioned on the clean speech.
Following the pix2pix’s architecture, we plan to implement U-Net as basic model of G and Revised PatchGAN as basic model of D. The conditional unit is feature map on speech spectrum.
ALGORITHM

1. Use the original data $D_{orig}$ to train an original recognition model $A$ and a cGAN model $N$ conditioned on clean data for data augmentation.
2. Use cGAN model $N$ to generate extra dataset $D_{gen}$.
3. Pool the original dataset $D_{orig}$ and generated data $D_{gen}$:
   A. with the same hard labels to train a new recognition model $B$
   B. use the model $A$ to get the soft labels for generated data and train a new recognition model $C$
EXPERIMENT

Figure 2: Visualization of real input, target and generated data

<table>
<thead>
<tr>
<th>model</th>
<th>data type</th>
<th>data size</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>vdcnn</td>
<td>baseline</td>
<td>original</td>
<td>9.10</td>
</tr>
<tr>
<td>vdcnn</td>
<td>synthesize</td>
<td>double</td>
<td>8.74</td>
</tr>
</tbody>
</table>