Motivation and Synopsys

Motivation: Given the scarcity of audio inputs and subtle differences in recording devices, systems that take audio as input must deal with poor quality audio in order to inform their actions.

Our contribution: We propose Audio Super Resolution Wasserstein GAN (ASRWGAN) to enhance the performance of ASRNet. Inspired by SWGAN, we utilize a pre-trained version of ASRNet as a generator with a fully convolutional discriminator inspired from WaveGAN.

Dataset Description and Preprocessing

- We use the CSTR VCTK dataset which includes 109 native English speakers each reciting 400 different English sentences. Due to compute restrictions, we train only on one speaker and sample half second patches for examples. This leads to the following split:
  - Train: 3328 examples, Validation: 500 examples.
- We preprocess each high resolution patch by using a Chebyshev low-pass filter to decimate the initial signal into a low resolution equivalent and then use bicubic interpolation for a baseline reconstruction.
- Generator receives the LR signals. The Discriminator receives both the initial HR signal (labeled real) and corresponding generator output PL signal (labeled fake).

Related Works


Final Infrastructure and Explanation

**Diagram Key**

- **Input:** Real Low Resolution Audio Signal
- **Output:** Final High Resolution Audio Signal

**Final Model Architecture (ASRWGAN)**

- **Model Details and Hyperparameters**
  - Discriminator Loss Function: $\mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] - \mathbb{E}_{z \sim p_{z}(z)}[\log (1 - D(G(z)))]$
  - Generator Loss Function: $\mathbb{E}_{x \sim p_{data}(x)}[\log G(x)]$
  - Learning Rate: 0.0001
  - Batch Size: 128
  - Leaky ReLU Activation with slope $\alpha = 0.01$
  - Shape of Generator Output: $1 \times 1 \times 64$
  - Weight Clipping Parameter: $c = 1$

- **System:**
  - Encoder-Decoder with a Residual Block
  - $\text{CONV1D}(n=256, \text{SAME})$
  - $\text{DROPOUT}(\text{alpha}=0.5)$
  - $\text{RELU}()$

- **Prediction:**
  - $\text{CONV1D}(n=128, \text{SAME})$
  - $\text{DROPOUT}(\text{alpha}=0.5)$
  - $\text{RELU}()$
  - $\text{DROPOUT}(\text{alpha}=0.5)$
  - $\text{PREDICT}()$

- **Diagram:**
  - Encoder-Decoder with a Residual Block
  - $\text{CONV1D}(n=256, \text{SAME})$
  - $\text{DROPOUT}(\text{alpha}=0.5)$
  - $\text{RELU}()$

Our Approach:

- Following SRGAN, combine pre-trained ASRNet (generator) with Discriminator inspired from WaveGAN.
- Replace Vanilla GAN with Wasserstein GAN with weight clipping to improve training stability
- Adapt Generator loss function to take into account a content component (MSE) to leverage super-resolution goal
- Add gradient clipping to prevent exploding losses for both the Generator and/or Discriminator.

Visual Results

Table 1: Objective metrics on audio super-resolution methods at an expansion ratio of 4

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR (dB)</th>
<th>SSIM</th>
<th>NIQE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASRNet</td>
<td>35.0</td>
<td>0.75</td>
<td>20.0</td>
</tr>
<tr>
<td>ASRWGAN</td>
<td>37.0</td>
<td>0.80</td>
<td>18.0</td>
</tr>
</tbody>
</table>

Table 2: Average MUSHRA test scores for each audio sample

<table>
<thead>
<tr>
<th>Condition</th>
<th>Sample 1</th>
<th>Sample 2</th>
<th>Sample 3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASRNet</td>
<td>70</td>
<td>65</td>
<td>75</td>
<td>70.3</td>
</tr>
<tr>
<td>ASRWGAN</td>
<td>72</td>
<td>68</td>
<td>73</td>
<td>70.3</td>
</tr>
</tbody>
</table>

Discussion

- In general, the ASRWGAN is strong at resolving the highest frequenices of the HR signal, especially when compared to the ASRNet.
- Performance can be boosted further by improving initial discriminator to leverage the value of a pre-trained generator.
- Balancing content loss and adversarial loss significantly affects performance.
- Overall, model successfully recovers and improves upon baseline performance over as few as 60 epochs.

Future Work

- Attempt to train the networks on multiple speakers in the VCTK dataset.
- Adapt model architecture further, specifically for the discriminator, by experimenting with pooling layers and adding skip connections and/or residual units.
- Modifying loss function, specifically the content loss portion, to more clearly encode strong audio signal reconstruction.
- Hyperparameter tuning for clipping bounds and integration of learning rate decay.
- Tune discriminator-to-generator training ratio.