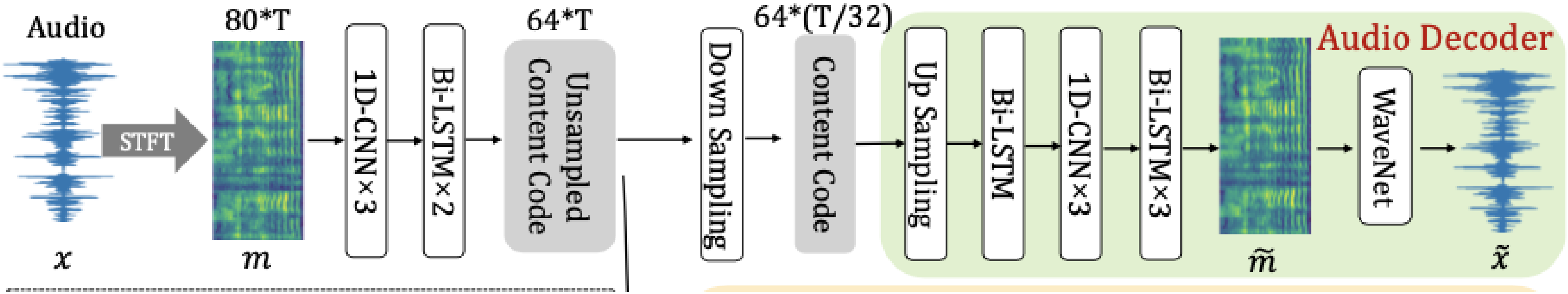
Cycle-Loss based Exemplar Autoencoder for Voice Conversion

Weida Liang

2021.11.10

Exemplar Autoencoder



Encoder **Decoder** **Vocoder**

Kangle Deng, Aayush Bansal, Deva Ramanan, “UNSUPERVISED AUDIOVISUAL SYNTHESIS VIA EXEMPLAR AUTOENCODERS“ in ICLR 2021

Compressibility of Audio Speech

- Speech contains two types of information: $x = f(s, w)$
  - (i) content(large variance)   (ii) style(little variance)

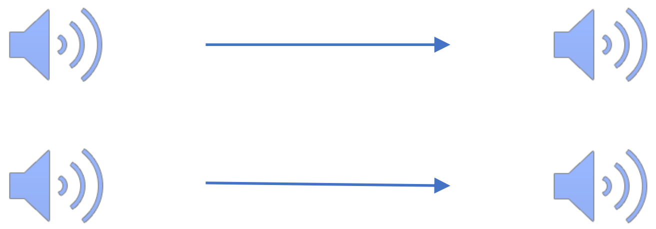
- Human Acoustics:
  - $Error\left(f(s_1, w_0), f(s_2, w_0)\right) \leq Error\left(f(s_1, w_0), f(s_2, w)\right), \forall w \in W$

- Autoencoder for Style Transfer:
  - $D\left(E(\hat{x})\right) \approx argMin_{t \in M} Error(t, \hat{x})= argMin_{t \in M} Error(t, f(s_1, w)) \approx f(s_2, w)$
    - M is the manifold spanning a particular style s_2.
  - Given sufficiently small bottlenecks, autoencoders can project out-of-sample points into the input subspace, so as to minimize the reconstruction error of the output.

Properties

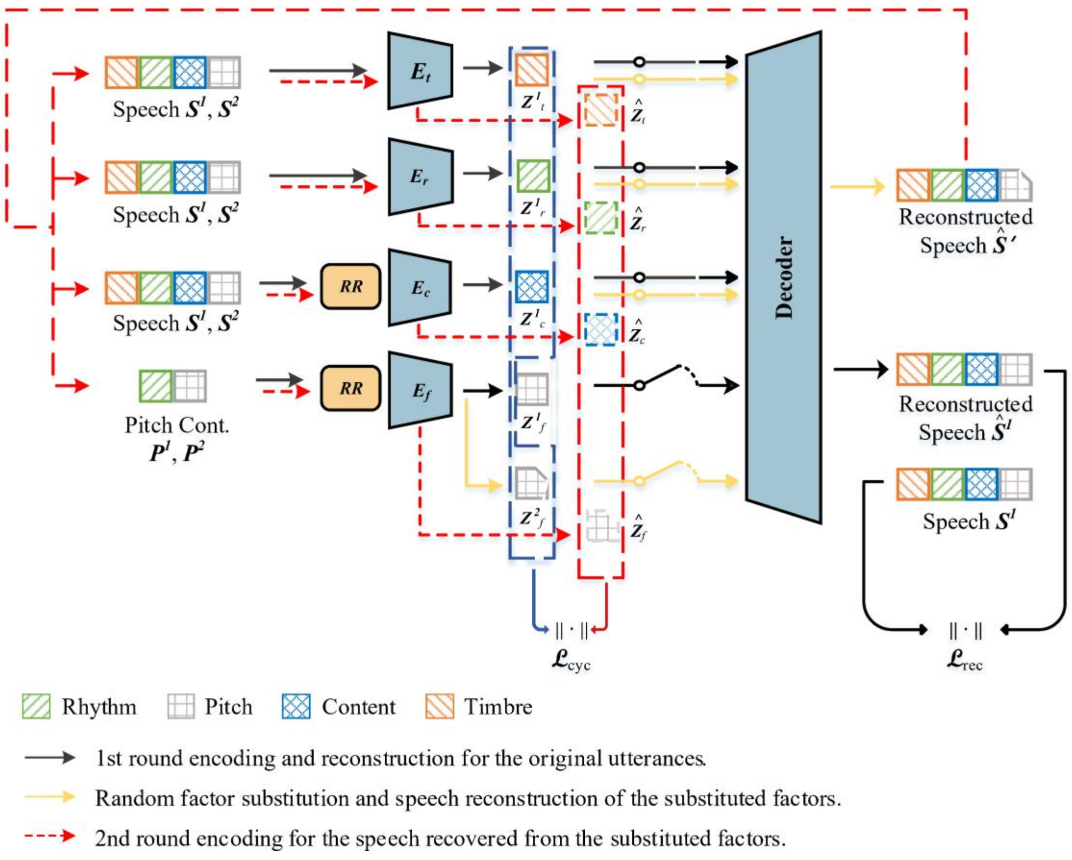
- Pros
  - A simple autoencoder framework(CNN+BI-LSTM)
  - Data-efficient and zero-shot
    - given a target speech with a particular, learn an autoencoder specific to that target speech

- Cons
  - Bad performance on cross-gender task
    - the content from the bottleneck and the speaker style from the weights are not purely factorized.



CycleFlow

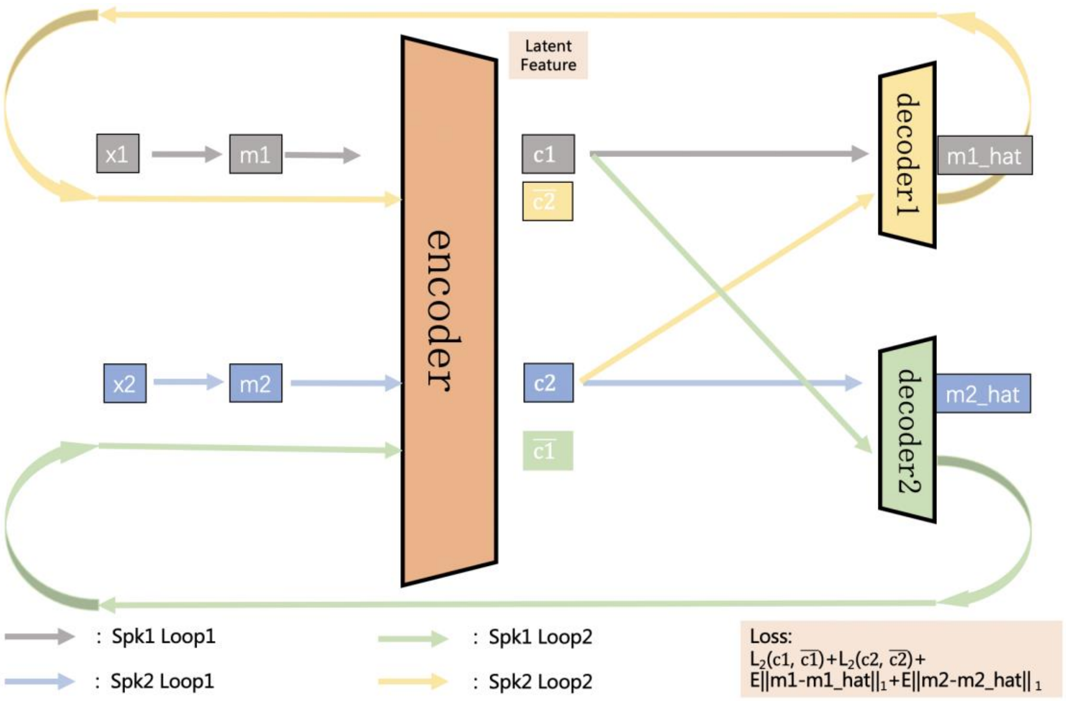
- **1st round encoding**: Firstly encode S^1 and S^2, resulting in two sets of factors: $\mathbf{Z}^1 = \{Z_r^1, Z_f^1, Z_c^1, Z_t^1\}$ and $\mathbf{Z}^2 = \{Z_r^2, Z_f^2, Z_c^2, Z_t^2\}$.
- **Random factor substitution (RFS)**: Randomly choose a factor from $\mathbf{Z}^2$, and use it to replace the corresponding factor in $\mathbf{Z}^1$. Suppose that the selected factor is Z_f^2 , we get a new factor set $\mathbf{Z}' = \{Z_r^1, Z_f^2, Z_c^1, Z_t^1\}$.
- **Speech reconstruction**: Forward Z' to the decoder and produce the reconstructed speech $\hat{S}'$.
- **2nd round encoding**: Encode $\hat{S}'$ and obtain $\hat{Z}' = \{\hat{Z}_r', \hat{Z}_f', \hat{Z}_c', \hat{Z}_t'\}$.
- The cycle loss is computed as: $\mathcal{L}_{cyc} = ||\boldsymbol{Z}' - \hat{\boldsymbol{Z}}'||^2$
- The final loss: $\mathcal{L} = \mathcal{L}_{rec} + \alpha * \mathcal{L}_{cyc}$



Haoran Sun, Chen Chen, Lantian Li, Dong Wang, "CYCLEFLOW: PURIFY INFORMATION FACTORS BY CYCLE LOSS "in ICASSP 2021

Cycle loss based Exemplar Encoder

- **1st round encoding**: Firstly convert x1 and x2 into spectrum m1 and m2; encode into latent space. Save latent features as c1 and c2.
- **Speech reconstruction**: Construct two decoders specific to speaker s1 and s2. Forward c1 and c2 to the decoder and produce the reconstructed spectrum m1_hat and m2_hat.
- **2nd round encoding**: Forward c1 and c2 separate to decoder2 and decoder1; then encode through common encoder again for latent features $\overline{c1}$ and $\overline{c2}$



Loss:

$$L_{cycle} = L_2(c1, \overline{c1}) + L_2(c2, \overline{c2})$$
$$L_{spec} = E\left||m1 - m1_{hat}|\right|_1 + E\left||m2 - m2_{hat}|\right|_1$$
$$L = \alpha * L_{cycle} + L_{spec}$$

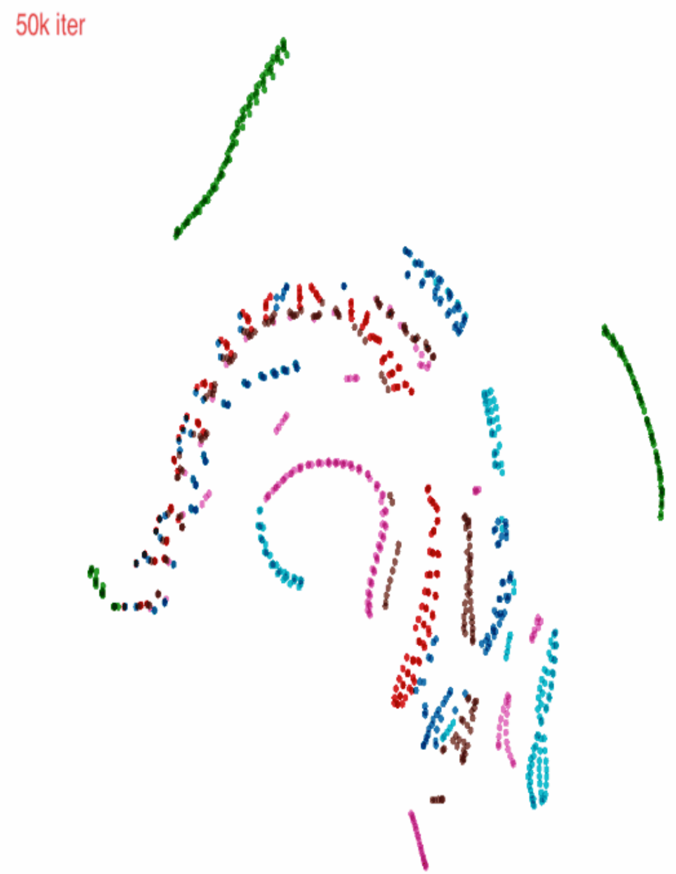
Check latent code to verify a best encoder

- We extract the content code from the output of the encoder and use this code for a further test.

- First, we choose six phones from the same speaker of the training period, each of which consists of 6 samples.

- Then set these phones as input into the autoencoder, and we can get the latent codes of these phones.

- Use tSNE to observe the clustering capibility of the phones. The dimension of the output of TSNE is 2.



Theoretical Analysis

- Define $x_1 = \{c_1, s_1\}$ for a speech of Spk1, where c_1 refers to content and s_1 refers to style. Same for Spk2.
- In an autoencoder, a reconstruction process refers to $D\big(E(x)\big)$
- For two encoders D_1 & D_2 specific for Spk1 and Spk2, further suppose $D_1\big(E(x_1)\big) = \widehat{x_1}$ for matched speech and decoder; $D_2\big(E(x_1)\big) = \overline{x_1}$ for mismatched speech and decoder.
- Then $||x_1 - \widehat{x_1}||^2 \rightarrow \big||E(x_1) - E(\widehat{x_1})\big||^2 = ||c_1 - \widehat{c_1}||^2 + ||s_1 - \widehat{s_1}||^2$,
  - $argmin_{\widehat{x_1}} \Big||x_1 - \widehat{x_1}||^2 = argmin_{\widehat{x_1}} \Big||D_1(E(x_1)) - \{c_1, s_1\}||^2 = \{c_1, \widehat{s_1}\}$. When training decoder1 with Spk1 speech, we have $\widehat{s_1} = s_1$, which means decoder1 has a manifold of s_1.
  - $argmin_{\widehat{x_2}} \Big||x_2 - \widehat{x_2}||^2 = argmin_{\widehat{x_2}} \Big||D_2(E(x_2)) - \{c_2, s_2\}||^2 = \{c_1, \widehat{s_2}\}$. When training decoder2 with Spk2 speech, we have $\widehat{s_2} = s_2$, which means decoder2 has a manifold of s_2.

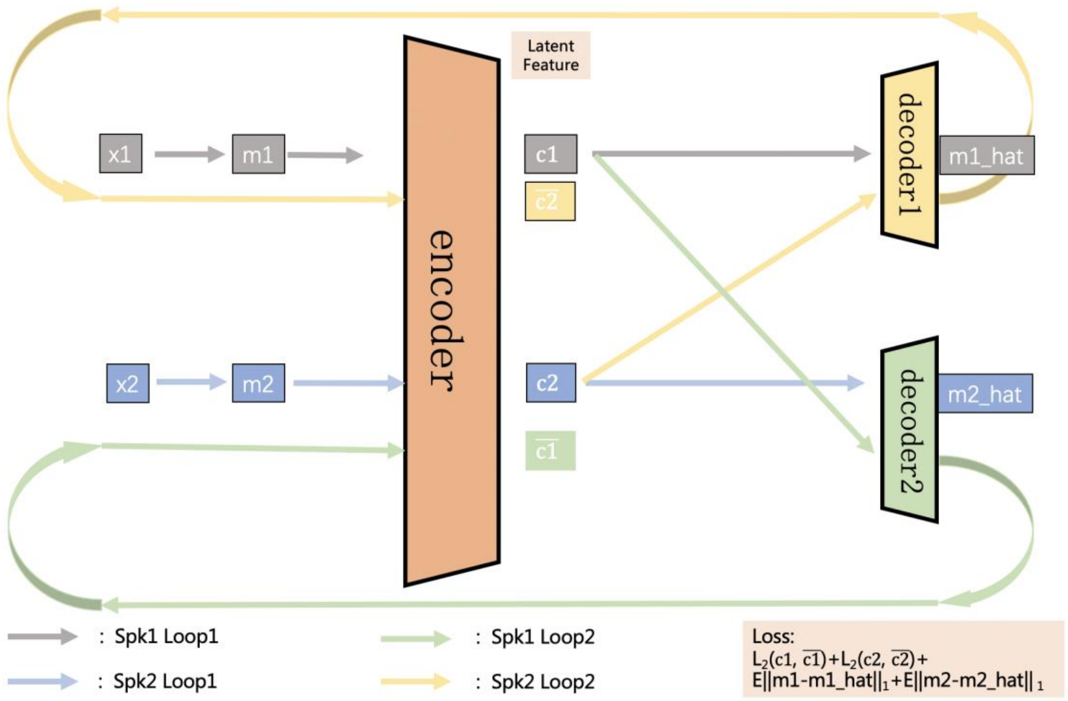
- While $||x_1 - \overline{x_1}||^2 \rightarrow \big||E(x_1) - E(\overline{x_1})\big||^2 = ||c_1 - \overline{c_1}||^2 + ||s_1 - \overline{s_1}||^2$
  - $argmin_{\overline{x_1}}\big||E(x_1) - E(\overline{x_1})\big||^2 = argmin_{\overline{x_1}}(||c_1 - \overline{c_1}||^2 + ||s_1 - \overline{s_1}||^2) = \{c_1, s_1\}$
  - With cycle loss, we are training a weaker decoder at a compensate for a stronger encoder .

Multi-Step Training

- **1st step**: Introduce cycle loss for a stronger encoder.

Loss:
$$L_{cycle} = L_2(c1, \overline{c1}) + L_2(c2, \overline{c2})$$
$$L_{spec} = E||m1 - m1_{hat}||_1 + E||m2 - m2_{hat}||_1$$
$$L = \alpha * L_{cycle} + L_{spec}$$

- **2nd step**: Fix the encoder and finetune the decoder for an autoencoder for a specific speaker.



Dataset and Configurations

- Training: A male speaker and a female speaker in AIShell dataset.
  - Speech length: 24:26(male) 26:53(female)
- Test: 6 speakers in AIShell dataset.
- The speakers and utterances in the training and test sets are not overlapped.
- Use TSNE to select a qualified encoder for decoder finetune.

Experiments

- 1. A comparison between not finetuned models with cycle loss and without cycle loss.

- 2. A comparison between decoder-finetuned models with cycle loss and without cycle loss.

Not Finetuned Models (With Griffinlim)

- Original Speech Baseline With Cycle Loss Without Cycle Loss

Conclusion1 : cycle-loss model does not have a better performance if not finetuned

Finetuned Models (With Wavenet)

- Original Speech Baseline With Cycle Loss Without Cycle Loss

Conclusion2 : cycle-loss model has a better performance if finetuned

Conclusion and Prospect

- 1. We proposed an improved autoencoder with multi-step training based on cycle loss.
- 2. We demonstrated theoretically and empirically that multi-step training has a better performance on cross-gender issue, while the model without finetune cannot reach that performance.
- 3. The proposed model preserved the advantage of simplicity in baseline.
- Future work:
  - Test for different IB dimensions.
  - Test for multi-step training with more speakers