One-shot Voice Conversion

治天锴

Background

- One/zero-shot ?
  - One-shot: few or even only one training sample, it can still make prediction
  - Zero-shot: There are no training samples for this class. But we can learn a mapping X -> y

- **Unsupervised VC**
  - Incorporate ASR system to perform unsupervised VC

    Shortage :**highly depend on the accuracy of the ASR system**

  - Utilize deep generative model like VAE or GAN

    Shortage: not synthesize the voice of the speakers who were never seen in training phase

Chou J, Yeh C, Lee H. One-shot voice conversion by separating speaker and content representations with instance normalization[J]. arXiv preprint arXiv:1904.05742, 2019.

Solution

- Assume : utterance = speaker representation + content representation

- Model

  - speaker encoder: encode the speaker information
  - content encoder: encode only the linguistic information
  - decoder: synthesize the voice back by combining these two representations

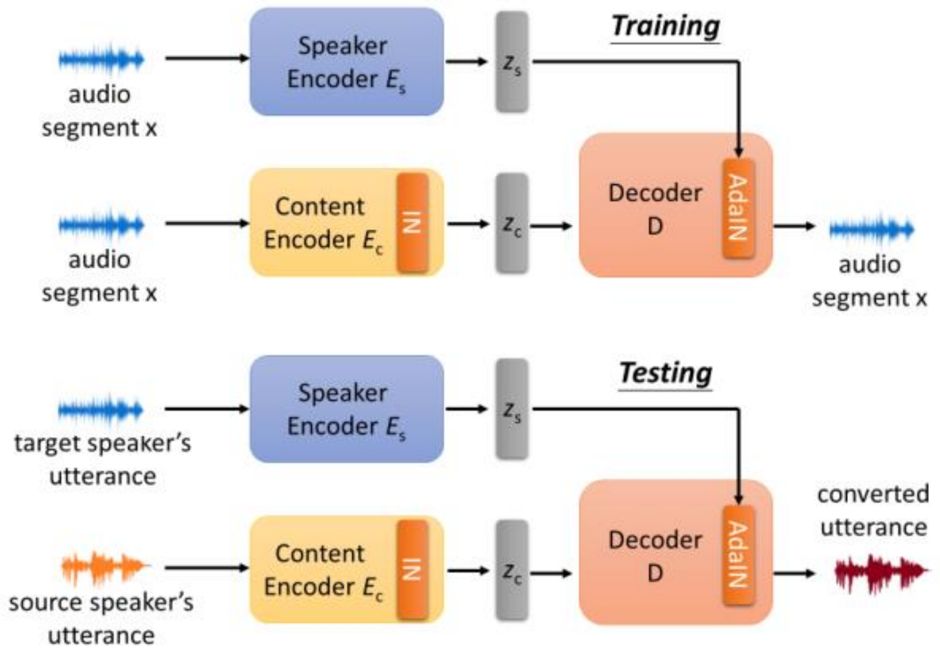

Figure 1: *Model overview.* $\mathrm{E_s}$ *is speaker encoder;* $\mathrm{E_c}$ *is content encoder and* D *is decoder. IN is instance normalization layer without affine transformation. AdaIN represents adaptive instance normalization layer.*

Details

- Variational autoencoder

$$L_{rec}(\theta_{\mathrm{E_s}}, \theta_{\mathrm{E_c}}, \theta_{\mathrm{D}}) = \mathop{\mathbb{E}}_{x \sim p(x), z_c \sim p(z_c|x)} [\| \mathrm{D}(\mathrm{E_s}(x), z_c) - x \|_1^1].$$

$$L_{kl}(\theta_{\mathrm{E_c}}) = \mathop{\mathbb{E}}_{x \sim p(x)} [\| \mathrm{E_c}(x)^2 \|_2^2].$$

$$\min_{\theta_{\mathrm{E_s}}, \theta_{\mathrm{E_c}}, \theta_{\mathrm{D}}} L(\theta_{\mathrm{E_s}}, \theta_{\mathrm{E_c}}, \theta_{\mathrm{D}}) = \lambda_{rec} L_{rec} + \lambda_{kl} L_{kl}$$

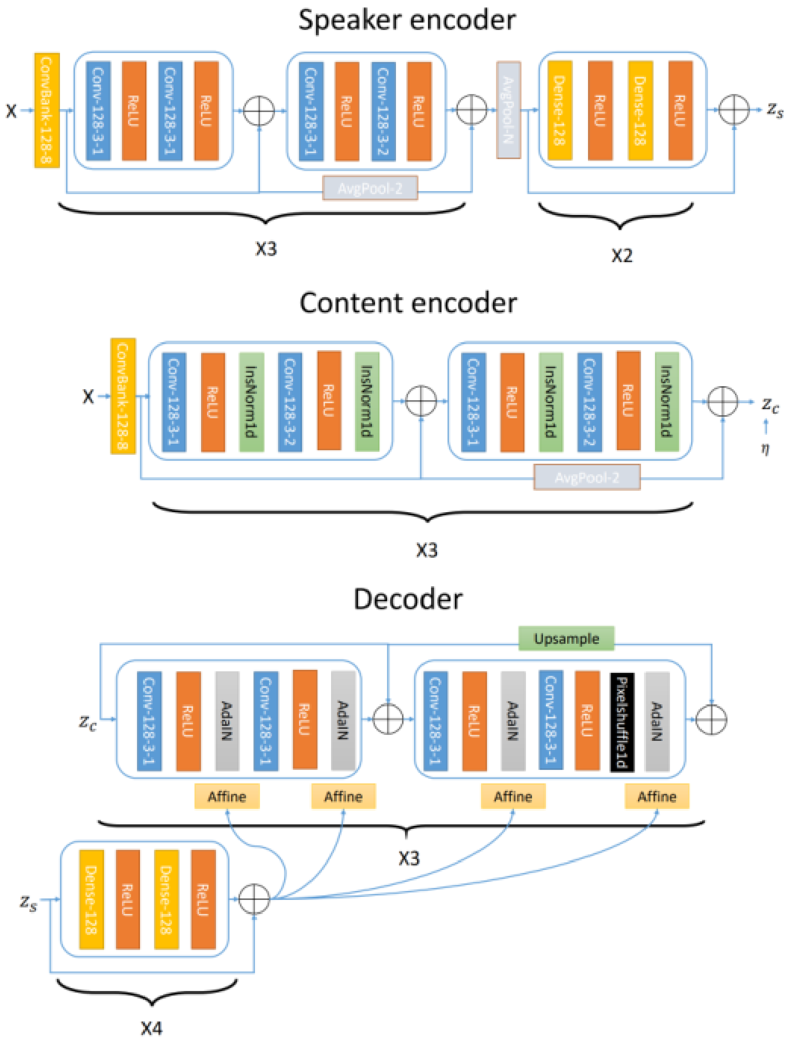

Figure 2: *The architecture of the encoders and decoder.*

Details

- Instance Normalization for Feature Disentanglement
  - instance normalization (IN) without affine transformation(remove the speaker information while preserving the content information)

$$\mu_c = \frac{1}{W}\sum_{w=1}^{W} M_c[w],$$

$$\sigma_c = \sqrt{\frac{1}{W}\sum_{w=1}^{W}(M_c[w]-\mu_c)^2+\epsilon},$$

$$M'_c[w] = \frac{M_c[w]-\mu_c}{\sigma_c}$$

  - enforce the speaker encoder to generate speaker representation(adaIN)

$$AdaIN(\mathbf{x},\mathbf{y}) = \sigma(\mathbf{y})(\frac{\mathbf{x}-\mu(\mathbf{x})}{\sigma(\mathbf{x})}) + \mu(\mathbf{y})$$

Table 1: *The accuracy for speaker identity prediction on content representation. Smaller value means less speaker information in the content representation.*

E_c **with IN**	E_c **w/o IN**	E_c **w/o IN +** E_s **with IN**
0.375	0.658	0.746

Experiments

- Dataset:CSTR VCTK Corpus(109 speakers)
  - Train:80 speakers
  - Validation:9 speakers
  - Test:randomly selected 20 speakers

https://zhitiankai.github.io/

AGAIN-VC System overview

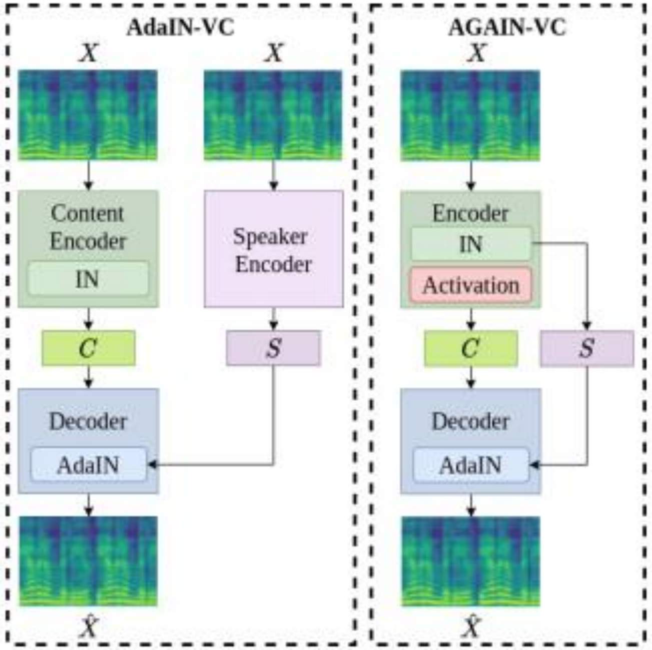

Fig. 1: AdaIN-VC and AGAIN-VC. AdaIN-VC uses a content encoder and a speaker encoder, while AGAIN-VC uses only one encoder and an activation to guide the training.

- AGAIN-VC: Activation Guidance and Adaptive Instance Normalization.

- With a proper activation as an information bottleneck on content embeddings

[2] Chen Y H , Wu D Y , Wu T H , et al. : A One-shot Voice Conversion using Activation Guidance and Adaptive Instance Normalization[J]. 2020.

Activation guidance (AG)

$$\text{Sigmoid}(x) = \frac{1}{1 + \exp(-\alpha x)},$$

Table 1: Comparison between the models with different activation functions. C and S are the speaker classification accuracy on content embeddings and speaker embeddings, respectively, and Rec. represents the reconstruction error. * is our proposed method.

Activation	C (Acc.%) ↓	S (Acc.%) ↑	Rec. ↓
None	68.6	92.6	0.161
ReLU	51.9	92.2	0.174
ELU	69.2	91.5	0.167
Tanh	57.3	91.7	0.165
Sigmoid ($\alpha = 1$)	30.5	90.0	0.167
Sigmoid ($\alpha = 0.1$) *	**1.7**	**93.2**	**0.151**
Sigmoid ($\alpha = 0.01$)	**1.7**	91.1	0.222

Table 2: Comparison between the models using a single encoder (1-Enc) and those with two encoders (2-Enc). Note that C and S represent the speaker classification accuracy on content embeddings and speaker embeddings, respectively; Rec. is the reconstruction error, and the last column is the model size. Also, "-sig" represents that sigmoid ($\alpha = 0.1$) is added on C; * is our proposed method.

	C (Acc.%) ↓	S (Acc.%) ↑	Rec. ↓	Size ↓
1-Enc	68.6	92.6	0.161	**9.5 M**
2-Enc	67.2	91.5	0.167	13.5 M
1-Enc-sig *	**1.7**	**93.2**	**0.151**	**9.5 M**
2-Enc-sig	1.8	92.7	0.154	13.5 M

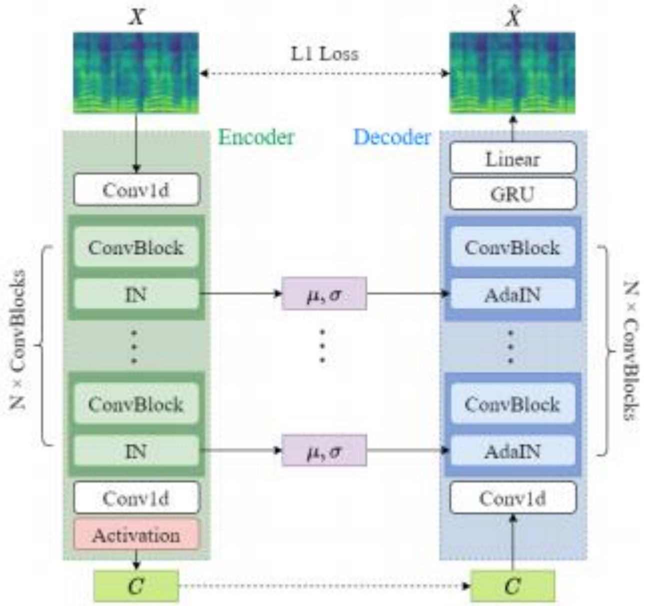

Fig. 2: AGAIN-VC architecture. The left part is the encoder, while the right part is the decoder. Note that L1 Loss is to make the input X and the output $\hat{X}$ as close as possible.

Thanks!