STARGAN-VC: NON-PARALLEL MANY-TO-MANY VOICE CONVERSION WITH STAR GENERATIVE ADVERSARIAL NETWORKS

Group Members:-

Rajat Kumar (1806040)

Pampana Bharath Kumar (1806044)

Gunturu Siddharth (1806131)

Authors :-

Hirokazu Kameoka

Takuhiro Kaneko

Kou Tanaka, Nobukatsu Hojo

NTT Communication Science Laboratories, NTT Corporation, Japan

Abstract

This paper proposes a method that allows non-parallel many-to-many voice conversion (VC) by using a variant of a generative adversarialnetwork (GAN) called StarGAN. Ourmethod, which we call StarGAN-VC, is noteworthy in thatit (1) requires no parallel utterances, transcriptions, or timealignment procedures for speech generator training, (2) si-multaneously learns many-to-many mappings across differ-entattribute domains using a single generator network, (3) is able to generate converted speech signals quickly enough to allow real-time implementations and (4) requires only severalminutes of training examples to generate reasonably realistic sounding speech. Subjective evaluation experiments on a non-parallel many-to-many speaker identity conversion task revealed that the proposed method obtained higher sound quality and speaker similarity than a state-of-the-art methodbased on variational autoencoding GANs.

Dataset

We use Voice Conversion Challenge (VCC) 2018 Dataset for our project which is used by authors also.The dataset consists of recordings of six female and six male US English speakers.We used a subset of speakers for training and evaluation.Specifically, we selected two female speakers ‘VCC2SF1’ and ‘VCC2SF2’. Two male speakers, ‘VCC2TM1’ and ‘VCC2TM2’.But in paper they used ‘VCC2SF1’, ‘VCC2SF2’, ‘VCC2SM1’ and ‘VCC2SM2’.There is nothing difference in that.

Preprocessing of Data

Before going to train the model it is required to pre-process the data.It is required to convert the audio samples into some featured vectors.For each utterance, a spectral envelope, a logarithmic fundamental frequency (log F0), and aperiodicities (APs) were extracted every 5 ms using the WORLD analyzer. A 36 Mel-cepstral coefficients (MCCs) were then extracted from each spectral envelope.Extracted features (mcep, f0, ap) from each speech clip are stored as npy files.We also calculate the statistical characteristics for each speaker.The extracted npy files contains an array with acoustic features theses arrays are used to train the model.

Training model - StarGAN-VC

While CycleGAN-VC allows the generation of natural-sounding speech when a sufficient number of training ex-amples are available, one limitation is that it only learnsone-to-one-mappings. Here, we propose using StarGAN [42] to develop a method that allows non-parallel many-to-many VC. We call the present method StarGAN-VC.

G be a generator that takes an acoustic feature sequence $x \in R\ Q \times N$ with an arbitrary attribute and a target attribute label c as the inputs and generates an acoustic feature sequence $y\hat{} = G(x, c)$.We assume that a speech attribute comprises one or more categories, each consisting of multiple classes. We thus represent c as a concatenation of one-hot vectors, each of which is filled with 1 at the index of a class in a certain category and with 0 everywhere else. For example, if we consider speaker identities as the only attribute category,c will be represented as a single one-hot vector, where each element is associated with a different speaker. One of the goals of StarGAN-VC is to make $\hat{y} = G(x, c)$ as realistic as real speech features and belong to attribute c.To realize this, we introduce a real/fake discriminator D as with CycleGAN and a domain classifier C, whose role is to predict to which classes an input belongs. D is designed to produce a proba-bility D(y, c) that an input y is a real speech feature whereas C is designed to produce class probabilities $p\ C\ (c|y)$ of y.

D is designed to produce a proba-bility D(y, c) that an input y is a real speech feature whereas C is designed to produce class probabilities p c (c|y) of y.

Adversarial Loss: First, we define

$$\begin{aligned}\mathcal{L}^{D}_{\text{adv}}(D) = &-\mathbb{E}_{c\sim p(c),\mathbf{y}\sim p(\mathbf{y}|c)}[\log D(\mathbf{y},c)]\\ &-\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),c\sim p(c)}[\log(1-D(G(\mathbf{x},c),c))],\\ \mathcal{L}^{G}_{\text{adv}}(G) = &-\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),c\sim p(c)}[\log D(G(\mathbf{x},c),c)],\end{aligned}$$

as adversarial losses for discriminator D and generator G, respectively, where y ∼ p(y|c) denotes a training example of an acoustic feature sequence of real speech with attribute c and x ∼ p(x) denotes that with an arbitrary attribute. $L_{adv}(D)$ takes a small value when D correctly classifies G(x, c) and y as fake and real speech features whereas L^{G}_{adv} (G) takes a small value when G successfully deceives D so that G(x, c) is misclassified as real speech features by D. Thus, we would like to minimize $L_{adv}(D)$ with respect to D and minimize L^{G}_{adv} (G) with respect to G.

Domain Classification Loss: Next, we define

$$\begin{aligned}\mathcal{L}^{C}_{\text{cls}}(C) &= -\mathbb{E}_{c\sim p(c),\mathbf{y}\sim p(\mathbf{y}|c)}[\log p_C(c|\mathbf{y})],\\ \mathcal{L}^{G}_{\text{cls}}(G) &= -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),c\sim p(c)}[\log p_C(c|G(\mathbf{x},c))],\end{aligned}$$

as domain classification losses for classifier C and generator G. L^{c}_{cls} (C) and L $^{c}_{cls}$ (G) take small values when C correctly classifies y ∼ p(y|c) and G(x, c) as belonging to attribute c. Thus, we would like to minimize L $^{C}{}_{\mathrm{cls}}$ (C) with respect to C and L $^{G}{}_{\mathrm{cls}}$ (G) with respect to G

Cycle Consistency Loss: Training G, D and C using only the losses presented above does not guarantee that G will pre-serve the linguistic information of input speech. To encourage G(x, c) to be a bijection, we introduce a cycle consistency loss to be minimized

$$\begin{aligned}&\mathcal{L}_{\mathrm{cyc}}(G)\\&=\mathbb{E}_{c'\sim p(c),\mathbf{x}\sim p(\mathbf{x}|c'),c\sim p(c)}[\|G(G(\mathbf{x},c),c')-\mathbf{x}\|_{\rho}],\end{aligned}$$

where x ∼ p(x|c ′) denotes a training example of an acoustic feature sequence of real speech with attribute c ′ and ρ is a positive constant. We also consider an identity mapping loss.

$$\mathcal{L}_{\mathrm{id}}(G)=\mathbb{E}_{c'\sim p(c),\mathbf{x}\sim p(\mathbf{x}|c')}[\|G(\mathbf{x},c')-\mathbf{x}\|_{\rho}],$$

to ensure that an input into G will remain unchanged when the input already belongs to the target attribute c ′ .

To summarize, the full objectives of StarGAN-VC to be minimized with respect to G, D and C are given as

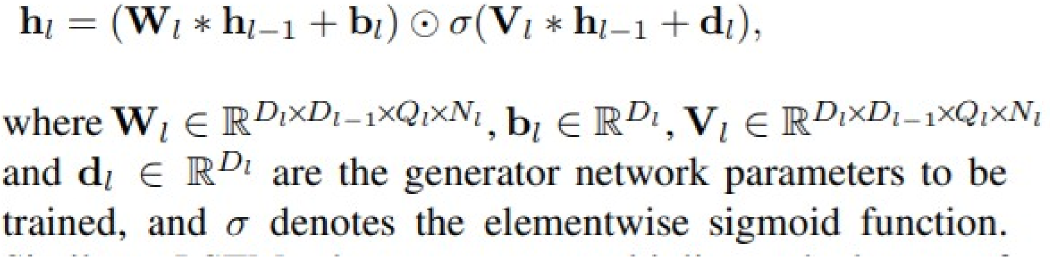
$$\begin{aligned}\mathcal{I}_G(G)=&\mathcal{L}_{\mathrm{adv}}^{G}(G)+\lambda_{\mathrm{cls}}\mathcal{L}_{\mathrm{cls}}^{G}(G)\\&+\lambda_{\mathrm{cyc}}\mathcal{L}_{\mathrm{cyc}}(G)+\lambda_{\mathrm{id}}\mathcal{L}_{\mathrm{id}}(G),\\\mathcal{I}_D(D)=&\mathcal{L}_{\mathrm{adv}}^{D}(D),\\\mathcal{I}_C(C)=&\mathcal{L}_{\mathrm{cls}}^{C}(C),\end{aligned}$$

respectively, where λ cls ≥ 0, λ cyc ≥ 0 and λ id ≥ 0 are regularization parameters, which weigh the importance of the domain classification loss, the cycle consistency loss and the identity mapping loss relative to the adversarial losses.

Convolutional Neural Networks

we treat an acoustic feature sequence x of size Q × N with 1 channel and use 2D CNNs to construct G

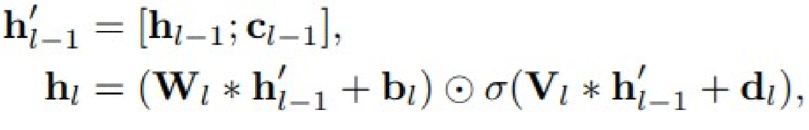
The output of the L-th hidden layer is described as a linear projection modulated by an output gate for encoder.



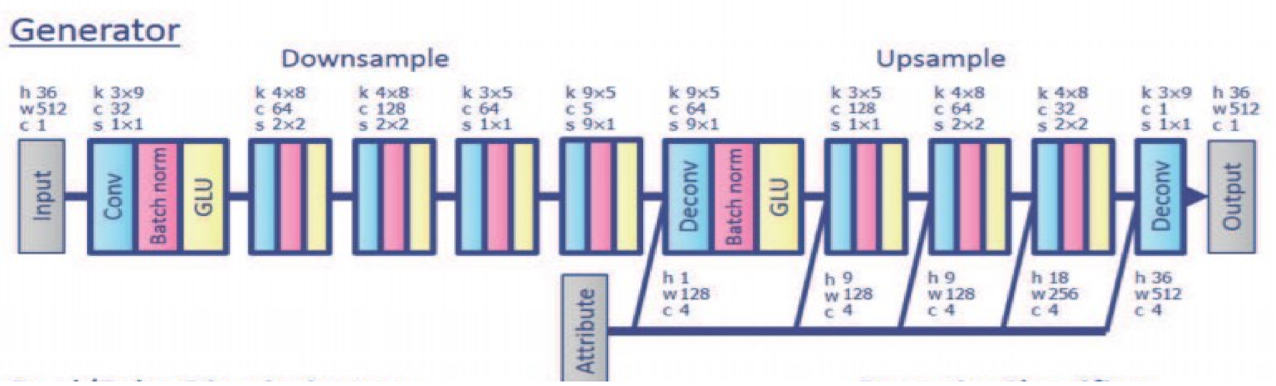
$$\mathbf{h}_l = (\mathbf{W}_l * \mathbf{h}_{l-1} + \mathbf{b}_l) \odot \sigma(\mathbf{V}_l * \mathbf{h}_{l-1} + \mathbf{d}_l),$$

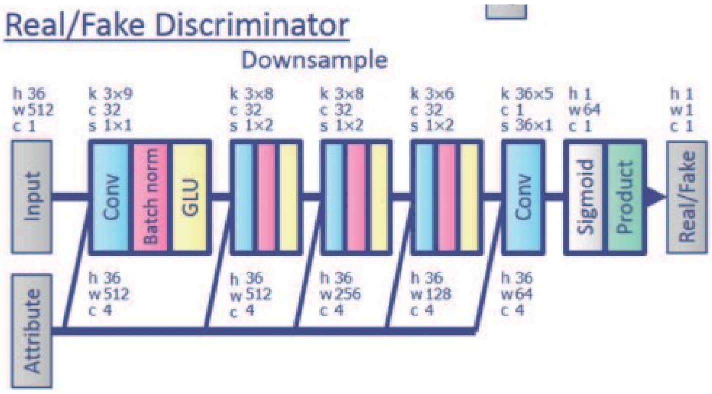
where $\mathbf{W}_l \in \mathbb{R}^{D_l \times D_{l-1} \times Q_l \times N_l}$, $\mathbf{b}_l \in \mathbb{R}^{D_l}$, $\mathbf{V}_l \in \mathbb{R}^{D_l \times D_{l-1} \times Q_l \times N_l}$ and $\mathbf{d}_l \in \mathbb{R}^{D_l}$ are the generator network parameters to be trained, and σ denotes the elementwise sigmoid function.

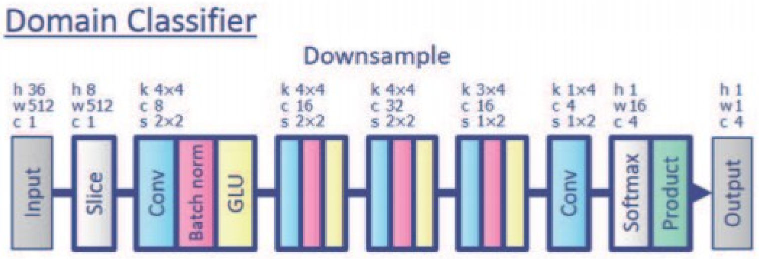
In the decoder part, the output of the L-th hidden layer is given by



$$\mathbf{h}'_{l-1} = [\mathbf{h}_{l-1}; \mathbf{c}_{l-1}],$$
$$\mathbf{h}_l = (\mathbf{W}_l * \mathbf{h}'_{l-1} + \mathbf{b}_l) \odot \sigma(\mathbf{V}_l * \mathbf{h}'_{l-1} + \mathbf{d}_l),$$







Result

Results given in paper

SM1 (Male) → SF1 (Female)

SF1 (Female) → SM1 (Male)

Results we got

TM1 (Male) → SF1 (Female)

SF1 (Female) → TM1 (Male)

As compare to the results given in paper our results are little bit noisy and some word are not clear.