ATTENTION-BASED WAVENET AUTOENCODER FOR UNIVERSAL VOICE CONVERSION

Adam Polyak[*] *Lior Wolf*

Facebook AI Research Tel Aviv University

ABSTRACT

We present a method for converting any voice to a target voice. The method is based on a WaveNet autoencoder, with the addition of a novel attention component that supports the modification of timing between the input and the output samples. Training the attention is done in an unsupervised way, by teaching the neural network to recover the original timing from an artificially modified one. Adding a generic voice robot, which we convert to the target voice, we present a robust Text To Speech pipeline that is able to train without any transcript. Our experiments show that the proposed method is able to recover the timing of the speaker and that the proposed pipeline provides a competitive Text To Speech method.

1. INTRODUCTION

In voice conversion, an audio sample from one person's voice is converted to the voice of another person. Ideally, the spoken words and the expression would remain intact by this transformation, but the identity of the speaker would change. Since the pace in which someone's speaks and the length of individual phonemes according to the context are an identifiable trait, the length of the sample after the conversion should not be forced to match the length of the input sample.

WaveNets [1] are powerful audio decoders that are able to map latent representations to high quality audio, including voice and music. It was recently shown that coupled with a dilated CNN encoder, music can be transformed into a latent semantic space and decoded back [2]. By training for multiple musical domains at once and adding domain adaptation terms, one can train a single encoder, which can transform any audio to a latent space that is shared by multiple music domains [3]. In this way, it is possible to obtain a "universal" conversion system, in the sense that the input can be from any music domain and the output is in one of the training domains.

In this work, we use a similar WaveNet-based encoder-decoder architecture for voice conversion. Our goal is to obtain a universal voice conversion network that, given any input voice, can convert it to one of the voices observed during training. However, WaveNet auto encoders are not suitable for this task, since they perform the conversion in a way that maintains the exact timing of the input. To tackle this problem, we introduce a new attention component that is able to learn the timing of each individual speaker.

Training the attention component requires having samples that are mis-aligned in time. This can be done using parallel corpora, of multiple speakers uttering the same sentences, which can be challenging to obtain. Instead, we train the attention mechanism by generating synthetic examples, in which the timing is randomly manipulated.

The entire voice conversion network contains three types of complex sub-networks, each dependent on the other types, and non-trivial to train: the voice encoder (a single dilated CNN), the voice decoder (one WaveNet per speaker), and the attention network (one per speaker). We tackle this challenge in several ways. First, the attention network is restricted to be a monotonic attention network, which combines ideas from the Gaussian attention network of Graves [4] and recent attention based networks [5]. Second, we devised a multi-phase training process, which defers the training of the attention module to later stages of the training.

Our voice conversion system, due to the universality property, has an immediate application in Text To Speech (TTS). We generate speech in a specific voice using an off-the-shelf voice robot and then convert its output to our target speakers. This pipeline is shown to be competitive to existing TTS methods, despite not involving the robot's voice in our training. In comparison to conventional TTS methods, we do not use transcribed samples for training our networks. Therefore, our proposed method can perform TTS based solely on unlabeled voice samples of an individual.

Our main contributions are: (i) demonstrating that a WaveNet autoencoder can be used for high quality voice conversion. (ii) presenting a method for a universal (no reference voice during training) voice conversion system. (iii) presenting a novel WaveNet autoencoder architecture that includes an attention mechanism, which allows for temporal fitting of the temporal speaking patterns of each of the speakers. (iv) presenting a WaveNet-based TTS method that does not require transcribed training data for new speakers.

Related work The applicability of a specific voice conversion method to a given task relies, among other factors, on the type and quantity of the required data. Our method requires no parallel data and does not require the reference (source) voice, unlike, e.g, [6]. However, due to the nature of the

[*]This work was carried out in partial fulfillment of the requirements for Adam Polyak Ph.D. degree.

WaveNet decoder, it requires considerable (unlabeled) data from the target domain. While this can be partly addressed by performing the conversion at a separate intermediate representation [7], in this work we opt for an end to end solution.

Methods that are based on parallel data can be less practical, since such data are not easy to obtain. An additional complication is that the methods often require alignment of the audio samples. A generic way to employ non-parallel data by iterating between nearest neighbor search of parallels and refinement of the conversion module, which in turn can improve matching, was proposed by Erro et al. [8]. While one can add a temporal component to this method, the transformation that is iteratively refined is mostly spectral and does not involve a change of the tempo. This is also true for the follow-up method [9], in which the transformation takes the form of adapting GMM coefficients.

In speaker verification, a GMM based representation, called i-vectors, is often used. By aligning the source and target GMMs, it is possible to achieve voice conversion in an unsupervised way [10]. The method employs an MFCC vocoder, which limits the output's quality. Here, too, the reference speakers are known and the tempo is unmodified.

While speaker verification features capture identity, speech recognition features capture the underlying text. In [11], a method was built which maps the input audio to the space of speech recognition features, where the two speakers are aligned. MFCC features are extracted from 25 ms windows, by employing a deep neural network, and the output audio is of the same length as the input audio.

Our method is based on a neural autoencoder. Previous works in the domain of voice conversion relied on variational auto encoders [12]. In [13], the basic idea of one encoder and a parameterized decoder, where the parameters represent the identity, was introduced. The method was subsequently improved [14] to include the WGAN [15]. In these work, spectral frames are used, while we use the waveform. In addition, a key component of our method is the temporal modification, which is not studied in these previous methods.

Neural TTS systems include the Deep Voice systems DV1 [16], DV2 [17] and DV3 [18], WaveNet [1] and Parallel WaveNet [19], Char2Wav [20], Tacotron [21] and Tacotron2 [22], and VoiceLoop [23] that was followed by VoiceLoop2 [24]. In our experiments, we compare to the open implementations of Tacotron2 and VoiceLoop. The former also employs a WaveNet decoder.

2. METHOD

The voice conversion method is based on training multiple autoencoder pathways, one per each of the k speakers, such that all paths share the encoder. During training, a softmax reconstruction loss is applied to the output of the WaveNet decoder in each domain separately. A domain confusion network [25] is trained to distinguish between the voice samples (classification into one of the k speakers), after the samples have been transformed to the latent space by the shared encoder. The encoder is trained adversarially, with respect to this network, in order to ensure that the encoding is not speaker-specific, thus enabling the universality property (converting from an unseen speaker). A diagram of the architecture is shown in Fig. 1(a). It includes the encoder E, the k decoders $D_1, \dots, D_k$, the domain confusion network C, and the attention networks $A_1, \dots, A_k$.

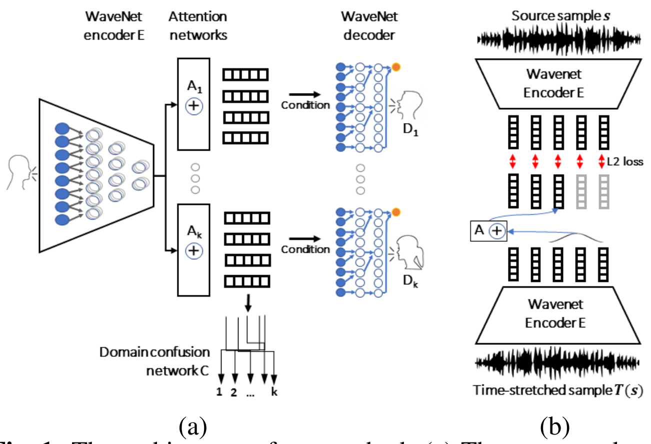

Fig. 1. The architecture of our method. (a) The autoencoder architecture and the confusion network. (b) The attention mechanism during training.

2.1. WaveNet Autoencoder

The architecture of the basic multi-speaker autoencoder follows [3]. It reuses the existing autoencoder architecture of [2], which is based on a WaveNet decoder and a WaveNet-like dilated convolution encoder. The k WaveNet decoders D_i, one per speaker, receive as input the sequence that was generated so far (an autoregressive model) and are conditioned on the latent representation produced by the encoder E.

We use a 16 kHz sample rate, which is quantized using 8-bit mu-law encoding, following [1, 2]. The shared encoder is a fully convolutional network with three blocks of 10 residual-layers each. Each residual-layer contains a ReLU nonlinearity, a dilated convolution with a gradually increasing kernel size, a second ReLU, and a 1×1 convolution layer followed by the residual summation of the activations before the first ReLU. We employ a fixed width of 128 channels throughout the blocks. After these, an additional 1×1 layer is applied, followed by an average pooling with a kernel size of 50 ms (800 samples). The obtained encoding is a vector in $\mathbb{R}^{48}$, which, due to the average pooling, is down-sampled by a factor of $\times 12.5$.

The conditioning signal is obtained by upsampling this encoding in the temporal domain to obtain the original audio rate and is applied to each of the WaveNet layers. It is passed through a 1×1 convolutional layer, that is different for each conditioning location. Each of the WaveNet decoders has four blocks of 10 residual-layers and a resulting receptive field of 250 ms (4,093 samples), as in [2]. At the top of each WaveNet, there are two fully connected layers and a softmax activation, which provides the probabilities of the quantized audio output (256 options) at the next timestep.

2.2. Attention Mechanism

We introduce a novel attention mechanism to the WaveNet autoencoder architecture above. The purpose of this module is to modify the samples temporally in order to capture the patterns of the target speaker. During training, the encoder E is used to encode an input signal s^j from speaker $j \in \{1 \dots k\}$, as well as a time-stretched version of the signal $T(s^j)$. The speaker-specific attention network A^j is then trained to recover the sequence of encodings of the original signal $E(s^j)$ from the time-augmented encodings $E(T(s^j))$, see Fig. 1(b).

The time augmentation operator T partitions the signal into segments of random i.i.d lengths of 0.3–0.5 sec., and time stretches each segment by a random factor of 50% to 150% of the original length, using the SoX audio processing tool.

We employ a GMM-based monotonic attention mechanism, which we found to be more stable than dot-product based attention mechanisms. The attention network A (omitting the index j for brevity) operates in an auto-regressive manner. It contains an LSTM, whose input is the concatenation of the previous time step of the signal, s_{t-1} (taken from the ground truth, i.e., teacher forcing is used), and the previously attended context, c_{t-1}, and predicts a vector in $\mathbb{R}^{3m}$ containing the prior γ_t of each of m GMM components, shifts of the centers of each Gaussian in the mixture κ_t, and log-variances β_t. A softmax is applied to the priors, in order to obtain a vector of pseudo-probabilities $\gamma'_t[i] = \frac{exp(\gamma_t[i])}{\sum_j exp(\gamma_t[j])}, i = 1, 2, \dots, m$. The means of the GMMs $\mu_t \in \mathbb{R}^m$ are increased $\mu_t = \mu_{t-1} + exp(\kappa_t)$, and the variances are computed as $\sigma_t^2 = exp(\beta_t)$. For each GMM component $1 \leq i \leq m$ and each point along the input sequence $1 \leq u \leq l$, we next compute $\phi[i, u] = \frac{\gamma'_t[i]}{\sqrt{2\pi\sigma_t^2[i]}} exp(-\frac{(u-\mu_t[i])^2}{2\sigma_t^2[i]})$. The attention weights α_t are then computed by summing across the components $\alpha_t[u] = \sum_{i=1}^m \phi[i, u]$, and the new context vector is given by multiplying the encoded input sequence $E(T(s))$ (viewed as a matrix with l columns), with the vector of weights $c_t = E(T(s))\alpha_t$. Finally, the predicted encodings at time t, is generated by passing the context vector c_t through a network of two fully connected layers.

For each speaker j, the attention mechanism is trained by minimizing the loss $\Psi_j = \sum_{s^j} \|(A^j(E(T(s^j))) - E(s^j)\|_F^2$, where A^j is the attention mechanism of speaker j, s^j is an input signal for speaker j, and the Frobenius norm considers the sequences as matrices with l columns.

2.3. Training and the Losses Used

The attention mechanism cannot be trained until the encoder is stabilized and we, therefore, perform training in two phases. Let s^j be an input sample in the voice of speaker $j \in \{1 \dots k\}$. Recall that E is the shared encoder, and D^j the WaveNet decoder for speaker j. Let C be the speaker classification network. It is comprised of three 1D-convolution layers, with the ELU [26] nonlinearity. The last layer projects an output vector of dimension k and is followed by a softmax. The CNN C is applied to samples of any length, and the output is averaged along the time axis to obtain a single $\mathbb{R}^k$ pseudo-probability vector.

The first phase trains the autoencoders (networks E, D^j), without applying the attention mechanism. The network C is applied to the output of the encoder $E(s^j)$ and is trained to predict j. During training, C minimizes the cross entropy classification loss $\Omega = \sum_j \sum_{s^j} \mathcal{L}(C(E(s^j)), j)$, and the autoencoders $j = 1, 2, \dots$ are trained with the loss $-\lambda\Omega + \sum_j \sum_{s^j} \mathcal{L}(D^j(E(s^j)), s^j)$, where $\mathcal{L}(o, y)$ is the cross entropy loss applied to each element of the output o and the corresponding element of the target y separately, and $\lambda = 10^{-2}$ is a weighing parameter. Note that the decoder D^j is an autoregressive model that is conditioned on the output of E. During training, the autoregressive model is fed the target output s^j from the previous time-step, instead of the generated output, a practice known as "teaching forcing". Unlike [3], no pitch augmentation is performed.

In the second phase of training, E is fixed, therefore C no longer plays a role, and the attention is introduced and trained jointly with the decoders using the loss ($\delta = 10^{-2}$ is a parameter) $\delta \sum_j \Psi_j + \sum_j \sum_{s^j} \mathcal{L}(D^j(A^j(E(s^j))), s^j)$.

Once the network is trained, given a new voice sample s to convert (existing or a new speaker), we apply the attenuated autoencoder of speaker j, in order to obtain a conversion to this voice: $D^j(A^j(E(s)))$.

2.4. Application to TTS

The method can be readily applied to obtain TTS in a given voice, by utilizing an existing TTS robot. First, a voice conversion system is trained with the target voice. There is no need to have samples of the underlying TTS robot, which means that the TTS robot can be easily replaced by future versions. Then, given text, the TTS robot generates a sample, which is converted to the target voice. Note that the process does not require transcribed samples of the target speaker. However, due to the nature of the WaveNet decoder, the (unlabeled) training set needs to be extensive.

3. EXPERIMENTS

Since we are unaware of public implementations of voice conversion systems, we focus our empirical validation on the application to TTS, which also seems to be a more competitive field (voice conversion work does not always compare to other state of the art methods). We do believe that our results in voice conversion are highly competitive, when considering the samples found online for various papers. However, owing to the data-hungry WaveNet decoder employed in our method, we were not able to perform such a comparison. Our samples are available online at `https://voiceconversion.github.io`.

We employed the following datasets, each with a single voice: (i) LJ [27], (ii) the Nancy corpus from the 2011 Blizzard Challenge [28], and (iii) the English audiobook data for the 2013 Blizzard Challenge [29]. Our method was compared to the ground truth (target speaker uttering the same sentence), as well as to Tacotron2 (using a NVIDIA's reimplementation [30] to which we added a WaveNet decoder) and VoiceLoop (using the authors' implementation). Our TTS samples were obtained by converting the Google Cloud TTS robot [31]. For our method, we present results with or without the attention mechanism, where the variant without attention is obtained after the first phase of training.

We use both automatic and user-study based success metrics: (i) The Mel Cepstral Distortion (MCD) scores. This is an automatic, albeit limited, method of testing the spectral compatibility between two audio sequences, in our case, the test sample of each dataset and the generated TTS sample of the corresponding text. Since the sequences are not aligned, MCD DTW, which uses dynamic time warping (DTW), is employed. (ii) In order to evaluate the temporal distance between the two sequences, we review the optimal warping achieved during the MCD DTW computation. Specifically, we count the total number of insertions and deletions. (iii) Mean Opinion Scores (MOS), on a scale between 1–5. These were computed using the "same_sentence" option of crowdMOS, following [17] (personal communication).

Tab. 1 presents MCD scores. In this metric, our method is on par with the baseline TTS methods, depending on the dataset. The MOS scores are shown in Tab. 2. As can be seen, our results are better than those of the other two TTS algorithms across datasets but do not match the score of the ground truth audio or the single speaker production-quality TTS robot. It is also apparent that the MOS of the method with the attention is somewhat lower than that of the MOS without applying the attention. However, it is still higher than the two literature methods.

The benefit of applying attention is apparent in Tab. 3, which shows a significant improvement in the temporal domain across all datasets in comparison to the no attention variant of our method. The temporal alignment of our method is also better than that of literature TTS methods.

The capability of the system to generate distinguished voices that match the original voices was tested, as was done in [17, 23, 24] using a speaker classifier. A multi-class CNN is trained on the ground-truth training set of multiple speakers, and tested on the generated ones. The network gets the inputs, after they were converted to the world vocoder features, performs five convolutional layers of 3×3 filters with 32 batch-normalized and ReLU activated channels, followed by max-pooling, average pooling over time, two fully-connected layers, and ending with a softmax of size three. The identification results are shown in Tab. 4. All methods seem to be highly identifiable, except, maybe, for Tacotron2 on LJ.

Table 1. MCD Scores (Mean ± SD; lower is better)

Method	LJ	Blizzard 2011	Blizzard 2013
Tacotron2	10.97 ± 0.54	8.57 ± 0.72	9.05 ± 1.37
VoiceLoop	13.55 ± 0.64	10.73 ± 0.86	10.18 ± 1.58
Our (no attn)	11.84 ± 0.51	8.75 ± 0.61	8.45 ± 0.41
Our	11.69 ± 0.41	9.38 ± 0.63	8.46 ± 0.37

Table 2. MOS Scores (Mean ± SD; higher is better)

Method	LJ	Blizzard 2011	Blizzard 2013
Tacotron2	2.63 ± 1.06	3.25 ± 0.98	2.57 ± 0.83
VoiceLoop	2.47 ± 1.07	3.21 ± 1.01	2.73 ± 1.21
Our (no attn)	3.78 ± 0.95	3.89 ± 0.76	3.49 ± 0.93
Our	3.50 ± 0.94	3.36 ± 0.92	3.21 ± 1.19
TTS Robot	4.35 ± 0.77	4.32 ± 0.75	4.25 ± 0.83
Ground truth	4.62 ± 0.68	4.72 ± 0.55	4.65 ± 0.63

Table 3. DTW distance results (Mean ± SD; lower is better)

Method	LJ	Blizzard 2011	Blizzard 2013
Tacotron2	4.20 ± 3.64	9.10 ± 4.25	22.37 ± 15.67
VoiceLoop	9.80 ± 4.15	10.18 ± 5.11	11.54 ± 6.18
Our (no attn)	2.74 ± 2.44	9.56 ± 4.63	7.67 ± 4.54
Our	1.76 ± 1.59	7.94 ± 3.21	5.28 ± 3.52

Table 4. Multi-Speaker Identification Top-1 Accuracy (%)

Method	LJ	Blizzard 2011	Blizzard 2013
Tacotron2	93.22	98.91	100.0
VoiceLoop	99.31	99.91	100.0
Our (no attn)	99.81	98.91	100.0
Our	99.80	98.95	100.0
Ground truth	100.0	99.74	100.0

4. CONCLUSIONS

A multi-speaker TTS system that requires language modeling for one speaker and untranscribed samples from all other speakers, has clear practical advantages, especially when the speakers can be gradually added over time and when the single speaker (the TTS robot) can be replaced without retraining the network. Here we propose a voice conversion method that is effective and flexible enough to enable this scheme: each target speaker is trained individually in an unsupervised manner, and conversion can be performed from any input voice.

From the algorithmic perspective, we are the first, as far as we know, to introduce an attention-based audio autoencoder that is applied directly to the waveform. This technology can be also useful, for example, in music conversion.

5. REFERENCES

[1] Aaron van den Oord et al., "Wavenet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[2] Jesse Engel et al., "Neural audio synthesis of musical notes with wavenet autoencoders," in *ICML*, 2017.

[3] Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman, "A universal music translation network," *arXiv preprint arXiv:1805.07848*, 2018.

[4] Alex Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[5] Ashish Vaswani et al., "Attention is All you Need," in *NIPS*. 2017.

[6] Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, Tomi Kinnunen, and Zhenhua Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," *arXiv preprint arXiv:1804.04262*, 2018.

[7] Li-Juan Liu, Zhen-Hua Ling, Yuan Jiang, Ming Zhou, and Li-Rong Dai, "Wavenet vocoder with limited training data for voice conversion," *Interspeech*, 2018.

[8] D. Erro, A. Moreno, and A. Bonafonte, "INCA algorithm for training voice conversion systems from nonparallel corpora," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, 2010.

[9] P Song, Y Jin, W Zheng, and L Zhao, "Text-independent voice conversion using speaker model alignment method from non-parallel speech," *INTERSPEECH*, 2014.

[10] T. Kinnunen, L. Juvela, P. Alku, and J. Yamagishi, "Non-parallel voice conversion using i-vector plda: towards unifying speaker verification and transformation," in *ICASSP*, 2017.

[11] Feng-Long Xie, Frank K. Soong, and Haifeng Li, "A KL divergence and DNN-based approach to voice conversion without parallel training sentences," in *INTERSPEECH*, 2016.

[12] Diederik P Kingma and Max Welling, "Auto-Encoding Variational Bayes," in *ICLR*, 2014.

[13] Chin-Cheng Hsu et al., "Voice Conversion from Non-parallel Corpora Using Variational Auto-Encoder," in *APSIPA*, 2016.

[14] Chin-Cheng Hsu et al., "Voice Conversion from Unaligned Corpora Using Variational Autoencoding Wasserstein Generative Adversarial Networks," in *INTERSPEECH*, 2017.

[15] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein Generative Adversarial Networks," in *ICML*, 2017.

[16] Sercan O Arik et al., "Deep voice: Real-time neural text-to-speech," in *ICML*, 2017.

[17] Sercan Arik et al., "Deep voice 2: Multi-speaker neural text-to-speech," in *NIPS*, 2017.

[18] Wei Ping et al., "Deep voice 3: 2000-speaker neural text-to-speech," in *ICLR*, 2018.

[19] Aäron van den Oord et al., "Parallel WaveNet: Fast high-fidelity speech synthesis," in *ICML*, 2018.

[20] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio, "Char2wav: End-to-end speech synthesis," in *ICLR workshop*, 2017.

[21] Yuxuan Wang et al., "Tacotron: A fully end-to-end text-to-speech synthesis model," in *INTERSPEECH*, 2017.

[22] Jonathan Shen et al., "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*, 2017.

[23] Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani, "VoiceLoop: Voice Fitting and Synthesis via a Phonological Loop," in *ICLR*, 2018.

[24] Eliya Nachmani, Adam Polyak, Yaniv Taigman, and Lior Wolf, "Fitting new speakers based on a short untranscribed sample," *ICML*, 2018.

[25] Yaroslav Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, 2016.

[26] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *ICLR*, 2017.

[27] Keith Ito, "The LJ speech dataset," 2017.

[28] Simon King and Vasilis Karaiskos, "The Blizzard challenge 2011," in *Blizzard Challenge workshop*, 2011.

[29] Simon King and Vasilis Karaiskos, "The Blizzard challenge 2013," in *Blizzard Challenge workshop*, 2013.

[30] Rahul Puri et al., "Tacotron 2 – pytorch implementation with faster-than-realtime inference," `https://github.com/NVIDIA/tacotron2`, 2018.

[31] Google Cloud TTS robot, "en-US-Wavenet-E," `https://cloud.google.com/text-to-speech/`, 2018.