Deep Voice Conversion

Shaoling Chen
New York University
Center for Data Science
sc6995@nyu.edu

Ruofan Wang
New York University
Center for Data Science
rw2268@nyu.edu

Kaitai Zhang
New York University
Center for Data Science
kz1179@nyu.edu

Abstract

Voice conversion represents converting one speaker's voice into another speaker's. It is an interesting topic and has many valuable use cases. In this project, we aims at building a voice conversion system using recurrent neural networks. We have two networks. The first one converts the source audio into a series of phones(phonemes) while the second one uses those phones to generate the target audio. It turns out our model works pretty well when converting from a speaker in TIMIT dataset to a U.S. female - the output audio is smooth, accurate, and natural.

1 Introduction

Voice Conversion(VC) is aiming at converting a source speaker's voice to another speaker's voice, while keeping the linguistic contents and characteristics. "In other words, VC modifies speaker-dependent characteristics of the speech signal in order to modify the perceived speaker's identity while keeping the speaker-independent information(linguistic contents) the same" (1). VC has many applications in real life, such as customized feedback of computer-aided pronunciation trimming systems, development of personalized speaking aids for speech-impaired subjects, automatic movie and animation dubbing (2).

A VC system is usually measured by the following three aspects: 1)How much linguistic information is retained in the converted voice? 2)How similar does the converted voice sound like the target speaker's voice? 3) How natural is the converted voice? Until now, building a voice conversion system that performs well on all three aspects is still an open and challenging problem.

There are two main approaches to build a VC system. The first one is to build an end-to-end system. This approach requires parallel training data, which is hard and expensive to get in real life. The second one is to use a combination of speech recognition and speech synthesis system. This approach doesn't require parallel training data and is much easier to implement. So in this project, we chose the second approach.

Our project focused on utilizing Deep Recurrent Neural Network techniques to build a combination of speech recognition and speech synthesis system to form our VC system. To be specific, in the speech recognition system, we extracted the MFCC feature sequence from the input voice and built a Bidirectional Long Short Term(bi-LSTM) Neural Network to predict the corresponding phone sequence. In the speech synthesis system, we built a Bidirectional Gated Recurrent Unit(bi-GRU) to transform the phone sequence to a magnitude spectrogram sequence and reconstructed the audio using Griffin-Lim Algorithm. We compared the performance of several common sequence-to-sequence network structures for both parts of the VC system and chose the best hyper-parameter setting for each structure. The results showed that our voice conversion system is able to capture important linguistic information of the sound of source speaker and sound naturally like the target speaker.

2 Related Work

Sun et al.(3) proposed a Bidirectional Long Short-Term Memory Neural Network (bi-LSTM)-based approach to build an end-to-end voice converison system. However, a parallel dataset is needed for building an end-to-end voice conversion system so we do not use his method. Later, Sun et al. (4) proposed a many-to-one voice conversion system without parallel data training. They first trained a bi-LSTM network to map MFCC features to Phonetic Posteriograms(PPG). Then they trained another bi-LSTM network to map the PPG to Mel-cepstral coefficients(MCEP) and used MCEP to synthesize the target speech. We use a same speech recognition system as their paper. However, in synthesizing the speech, we choose to map the PPG to spectral magnitude on a linear scale because it is simpler to implement. Also, our choice of such a speech synthesis system is validated by The Tactron: Towards End-to-End Speech Synthesis (5) paper by Google.

There are also other ways to build voice conversion system without using parallel data, such as using variational auto encoders(VAE) to model the latent structure of speech in an unsupervised manner(6). Due to the limitation of time, we do not use their approach and these ways can be explored in the future work.

3 Problem definition and algorithms

3.1 Task

Our task is to convert a source speaker's voice(it can be someone from the training set or some arbitrary character, like Peppa Pig) to a US female's voice(target speaker). The core of our work is to build a speech recognition system that transforms the input, the source speaker's speech into phone sequences and another speech synthesis system that converts phone sequences into the output speech. The performance of a voice conversion system is measured by the output sound's similarity to the target speaker, its naturalness and the proportion of retained linguistic information.

3.2 Algorithm

3.2.1 MFCC Feature Extraction

The speech signal is considered as stationary over a short time period(<100 ms) but non-stationary over a long time range due to different sounds being uttered. Therefore, we can use a short-time spectral analysis to capture the feature of the input speech. In our project, we use Mel-Frequency Cepstrum Coefficients(MFCC) features to capture the short-time spectral pattern of a speech. The process of MFCC feature extraction is divided into the following five steps.

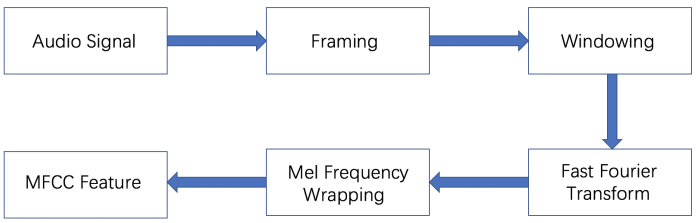

Figure 1: MFCC Feature Extraction Process

Framing In the first step, the speech signal is framed into samples.We set the sample frame length as 25ms, the shift frame as 5 ms in our project.

Windowing If the signal is discontinuous between two ends, the extracted spectrum will be distorted. Therefore, a windowing function is applied to each sample frame, which connects the beginning and the end of each segment. Hamming window function is a commonly used windowing function and is used in our project.

Fast Fourier Transform(FFT) Fast Fourier Transform is used to get the spectrum of the signal. We set the sampling frequency as 512 Hz.

Mel-frequency Wrapping Since a mel scale(a linear spacing below 1000 Hz and a logarithmic spacing above 1000 Hz) is more similar to the human perception of the speech's frequency information(7) , we convert the spectrogram we get from FFT to a mel-spectrogram in this step.

Mel-Frequency Cepstrum Coefficients(MFCC) In the final step, we use Discrete Cosine Transform(DCT) to convert the log mel spectrum back to time. The result is also called the mel frequency cepstrum coefficients (MFCC), which provide a good representation of the signal's local spectral properties(7). In our project, a 40-dimension MFCC feature vector is extracted for each time frame.

3.2.2 Phones(Phonemes)

A phone(phoneme) is a speech segment possessing distinct physical or perceptual properties, which serves as the basic unit of phonetic speech analysis (11). The visual representation of phones is called phonetic transcription. For example, the word 'she' has two phones [sh] and [ey]. In our work, TIMIT original transcriptions(12), which are based on 61 phones, are used(Figure2).

	Phone Label	Example		Phone Label	Example		Phone Label	Example
1	iy	b*ee*t	22	ch	*ch*oke	43	en	butt*on*
2	ih	b*i*t	23	b	*b*ee	44	eng	Wash*ing*ton
3	eh	b*e*t	24	d	*d*ay	45	l	*l*ay
4	ey	b*ai*t	25	g	*g*ay	46	r	*r*ay
5	ae	b*a*t	26	p	*p*ea	47	w	*w*ay
6	aa	b*o*b	27	t	*t*ea	48	y	*y*acht
7	aw	b*ou*t	28	k	*k*ey	49	hh	*h*ay
8	ay	b*i*te	29	dx	mu*dd*y	50	hv	a*h*ead
9	ah	b*u*t	30	s	*s*ea	51	el	bott*le*
10	ao	b*ough*t	31	sh	*sh*e	52	bcl	b closure
11	oy	b*oy*	32	z	*z*one	53	dcl	d closure
12	ow	b*oa*t	33	zh	a*z*ure	54	gcl	g closure
13	uh	b*oo*k	34	f	*f*in	55	pcl	p closure
14	uw	b*oo*t	35	th	*th*in	56	tcl	t closure
15	ux	t*oo*t	36	v	*v*an	57	kcl	k closure
16	er	b*ir*d	37	dh	*th*en	58	q	glotal stop
17	ax	*a*bout	38	m	*m*om	59	pau	pause
18	ix	debit	39	n	*n*oon	60	epi	epenthetic silence
19	axr	butt*er*	40	ng	si*ng*			
20	ax-h	*su*spect	41	em	bott*om*	61	h#	begin/end marker
21	jh	*j*oke	42	nx	wi*nn*er			

Figure 2: Phone Set

3.2.3 Recurrent Neural Network

In order to model sequential data(audio features), we use recurrent neural network(RNN)(13) in this case. Recurrent neural network is a powerful tool since it incorporates the 'recursive' idea in computer science, so that it can efficiently capture sequential information.

The most basic RNN takes the current observation and the previous one hidden state as the input, outputs the hidden state in the current stage and passes it to the next cell. The problem is that it may suffer from gradient explosion and gradient vanishing problem. To solve this problem, several other versions of RNN architecture have been proposed, including GRU and LSTM.

Long Short Term Memory(LSTM)(14) network uses three gates to control information flow: input gate, forget gate, and output gate. Each gate is a nonlinear unit that takes activation from the input data and hidden state. Input gate determines which part of the newly computed information should be kept, and forget gate determines which part of information from the last cell should be neglected. Output gate determines which part of the current information should be exposed to the external

network. By adding some shortcuts between time steps(as shown in Figure 3), LSTM solved gradient vanishing problem.

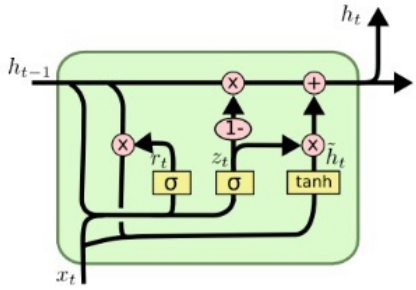

Figure 3: LSTM cell architecture

Gated Recurrent Units(GRU)(15) is another modified version of RNN. The basic idea is similar to that of LSTM, but it combines the input gate and the forget gate into a single gate called the update gate(Figure 4). In this way, it saves a lot of computation as well as maintaining good performance.

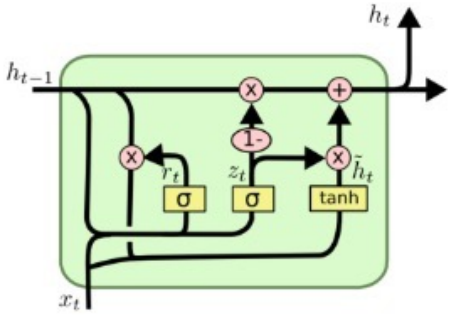

Figure 4: GRU cell architecture

A main problem with uni-directional RNNs is it can only learn hidden representations from the previous observation or time step, but this is not always what we want. In audio data or natural language, most of the time we need to look at the surrounding to help us understand the context. Therefore, bi-directional RNNs(16) are proposed. In the forward propagation, we firstly move from left to right until we come to the last time step, then we move from right to left until we reach the inital point(Figure 5).

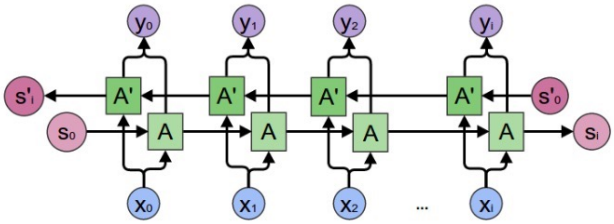

Figure 5: Bi-RNN architecture

In our work, we implemented both uni-directional and bi-directional LSTM and GRU on our data, and the results and comparison will be covered in the next sections.

3.2.4 Griffin-Lim Algorithm

We use the Griffin-Lim algorithm to synthesize waveform from the magnitude spectrum, the log magnitude of the short-time fourier transform of the audio signal. The iterative algorithm attempts to decrease the mean squared error between the STFT magnitude of the estimated signal and the modified STFT magnitude. It randomly initializes a phase spectrum and then in every iteration, computes the inverse-STFT and re-estimates the phase of the STFT(17).

4 Experiments

4.1 Data

The dataset we use to build the speech recognition system is the TIMIT Speech Corpus. "It is large enough to be a reasonable acoustic modeling benchmark for speech recognition, yet it is small enough to keep a large study such as ours manageable"(8). The TIMIT dataset consists of utterances of 630 speakers, of phonetically rich sentences. There are a total of 4620 files constituting the training dataset and 1680 files constituting the test dataset. The wav files are sampled at 16000 Hz and are of varying lengths. We also have the corresponding phones along with their start and end times in the audio. These are converted to per frame phones where all the frames between the start and end time of a phone are labelled the same. A one hot representation of these phones forms the output of the network.

The dataset we use to build the speech synthesis system is the CMU Arctic dataset(10). This dataset has 593 train and 539 test samples for each individual speaker of varying accents and genders, which makes it a good dataset for training target voices. The phone labels along with the end time (in secs) for each are available in the transcribed label files. We trained our model only on all the utterances of one American female speaker with no accent.

For the combination of these two systems, we need to check their consistency of phone labels. We find while the TIMIT dataset uses the label ‘h’ for pauses, the Arctic dataset uses the label ‘pau’ . Thus we have to convert the ‘pau’ to ‘h’ in the Arctic dataset.

4.2 Methodology

4.2.1 Model Architecture

As mentioned in the previous section, we used two separate neural networks in our work. The optimal model architecture is selected based on our experimental results, which will be covered in the next sub-section.

The first network (Figure6) takes the audio series as the input, converts it to MFCC features, then feeds the features to a bi-directional RNN to get a series of phones(phomemes).

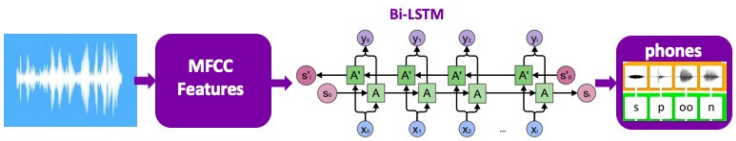

Figure 6: Net 1 Architecture

The second network (Figure7) takes the output from the first network as an input and uses another bi-directional RNN to generate the magnitude spectrum. Then we use the Griffin-Lim algorithm to convert the magnitude spectrum we get to audio signals.

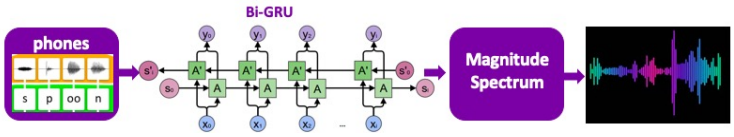

Figure 7: Net 2 Architecture

4.2.2 Evaluation

In the first network, we use a cross-entropy loss as the loss function. The accuracy is calculated by the percentage of phones correctly classified. In the second network, we use the mean squared error between the predicted and actual magnitude spectrum values as the loss metric. In both networks, the Adam Optimizer is used to minimize the loss.

4.3 Experiments and Results

4.3.1 Net 1

Firstly, we tried different model architectures(uni-GRU, bi-GRU, uni-LSTM, bi-LSTM) with the same hyperparameters. The validation accuracy is listed in the table 1:

Model Arch.	Hidden Size	#Hidden Layers	Dropout	Val. Acc
Uni-LSTM	100	2	0.4	0.685
Bi-LSTM	100	2	0.4	0.690
Uni-GRU	100	2	0.4	0.684
Bi-GRU	100	2	0.4	0.688

Table 1: Net 1: model selection

From these results we can see, under the same configurations, bi-directional LSTM performs best among the four models. So we choose bi-LSTM as our model framework and do hyperparameter searching to finalize the configuration.

For hyperparameters, we focused on hidden size, the number of hidden layers, and the dropout rate. The detailed configurations we have tried are listed in the table 2. Due to the page limit, we won't list all the validation accuracy. The best configuration among them is: hidden size = 200, the number of layers = 3, and dropout rate = 0.4.

Hidden Layer	Hidden Size	Dropout
2	[100,200]	[0.4,0.3,0.2]
3	[100,200]	[0.4,0.3,0.2]
4	[100,200]	[0.4,0.3,0.2]

Table 2: Net 1: hyperparmeter tuning

Under the best configuration for net1, the training and validation curve is plotted below in Figure 8. As we can see, the best validation accuracy achieved 0.721. Thus we finalized our net 1.

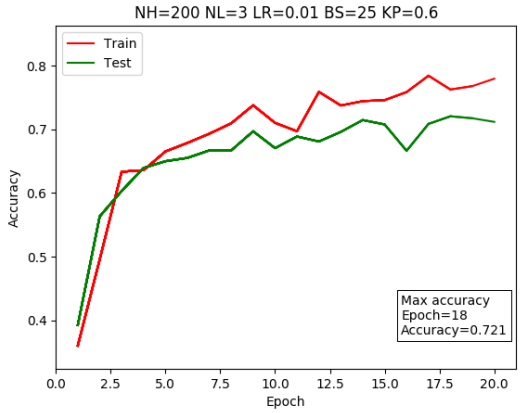

Figure 8: Net 1: best configuration

4.3.2 Net 2

In net 2, we adopted the same model selection apporoach as network 1, where we firstly picked the best model framework, then conducted hyperparameter tuning. The only difference is that instead of using validation accuracy to evaluate, we use MSE(mean squared error) in this case.

In the model selection part, we have the following results shown in table3.

Model Arch.	Hidden Size	#Hidden Layers	Dropout	Val. MSE
Uni-LSTM	100	2	0.1	0.206
Bi-LSTM	100	2	0.1	0.206
Uni-GRU	100	2	0.1	0.208
Bi-GRU	100	2	0.1	0.205

Table 3: Net 2: model selection

From these results we decided to use Bi-GRU in this network, for the hyperparameter tuning, we tried the following configurations(Table 4)as well:

Hidden Layer	Hidden Size	Dropout
2	[100,200]	[0.3,0.2,0.1]
3	[100,200]	[0.3,0.2,0.1]
4	[100,200]	[0.3,0.2,0.1]

Table 4: Net 2: hyperparmeter tuning

The finalized structure for net 2 is: bi-directional GRU with hidden size = 100, the number of hidden layers = 4, and dropout rate = 0.1. Under this configuration, the training and validation curve is plotted in Figure9.

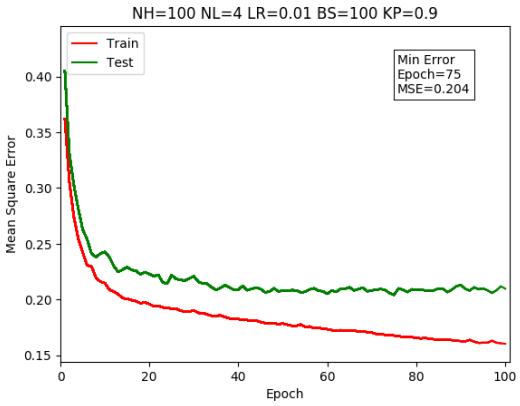

Figure 9: Net 2: best configuration

4.4 Discussion

We convert an audio from the TIMIT dataset to the target voice from the Arctic dataset successfully and the converted waveform retains most features of the target speaker, which means that our model works pretty well for the designated source speaker(from TIMIT dataset) and the target speaker(the given U.S. female).

However, when we try to use Peppa Pig's voice as source audio, the conversion result is not that good. Although it still retains the intelligibility of the spoken sentence, the voice doesn't sound that natural. The reason is that the dataset used in Net 1 does not reflect the audio characteristics of Peppa Pig's successfully. In order to convert Peppa Pig's voice to others, we need sufficient Peppa Pig's audio data as well as the corresponding MFCC features.

Here is an example from the Peppa Pig that is converted to the target voice: Original Voice, Target Voice, Converted Voice.

5 Conclusion and Future Work

5.1 Conclusion

- Recurrent neural networks work pretty well on audio data since it can efficiently capture the sequential features, this is also why RNN has become the state-of-art method for natural language processing and time series modelling.
- Our model converts a source speaker's voice from the TIMIT dataset to the target speaker's voice perfectly, but does not perform that well if we change the source speaker to Peppa Pig, which means that sufficient source data and feature extraction is critical to build a robust model.

5.2 Future Work

Our model successfully converts the source speaker's voice to the target speaker's voice, keeping most of the linguistic information. For the limitation of the output sound's similarity to the target speaker, we would like to collect more Peppa Pig audios to construct customized dataset and focus on increasing the performance of Net 2. One direction is to explore deeper models like Tactron to get better results.

6 Contribution

Shaoling Chen: built the speech recognition system and tested models for Net 1; drafted final write-up

Ruofan Wang: built the speech synthesis system and tested models for Net 2; drafted final write-up

Kaitai Zhang: preprocessed the TIMIT and Arctic datasets; tested audio reconstruction methods; drafted final write-up

References

[1] Seyed Hamidreza Mohammadi, Alexander Kai, An overview of voice conversion systems.Speech Communication, Volume 88, April 2017, P 65-82

[2] Lifa Sun: Real-time Voice Conversion with Phonetic Posteriorgrams
`http://www1.se.cuhk.edu.hk/ lfsun/vc`

[3] Lifa Sun, Shiyin Kang, Kun Li, Helen Meng, Voice Conversion Using Deep Bidirectional Long Short-Term Memory Based Recurrent Neural Networks.In Proc. ICASSP, 2015.

[4] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang and Helen Meng, Phonetic Posteriograms for Many-to-one Conversion Without Parallel Data Training. In the Proceedings of IEEE International Conference on Multimedia and Expo, ICME 2016

[5] YuxuanWang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weissy , Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengioy, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, Rif A. Saurous, Tactron: Towards End-to-end Speech Synthesis. In Proceedings of Interspeech, August 2017a. URL:https: //arxiv.org/abs/1703.10135.

[6] Wen-Chin Huang, Hsin-Te Hwang, Yu-Huai Peng, Yu Tsao, Hsin-Min Wang, Voice Conversion Based on Cross-Domain Features Using Variational Auto Encoders. https://arxiv.org/abs/1808.09634

[7] Ch.Srinivasa Kumar, P. Mallikarjuna Rao, Design Of An Automatic Speaker Recognition System Using MFCC, Vector Quantization And LBG Algorithm.In International Journal on Computer Science and Engineering, Vol. 3, no. 8, 2011.

[8] JS Garofolo, LF Lamel, WM Fisher, JG Fiscus, DS Pallett, and NL Dahlgren. DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM. National Institute of Standards and Technology, NTIS Order No PB91- 505065, 1993

[9] Klaus Greff, Rupesh K. Srivastava, Jan Koutnık, Bas R. Steunebrink, Jurgen Schmidhuber, LSTM: A Search Space Odyssey. IEEE transactions on neural networks and learning systems, 2017.

[10] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in Fifth ISCA Workshop on Speech Synthesis, 2004.

[11] Phone (phonetics)
`https://en.wikipedia.org/wiki/Phone(phonetics)`

[12] Zue, V. Seneff, S, Transcription and alignment of the TIMIT database. In Hiroya Fujisaki (Ed.), Recent research toward advanced man-machine interface through spoken language. Amsterdam: Elsevier, pp 464-447, 1996.

[13] Lipton, Zachary C., John Berkowitz, and Charles Elkan. "A Critical Review of Recurrent Neural Networks for Sequence Learning." ArXiv:1506.00019 [Cs], May 29, 2015. http://arxiv.org/abs/1506.00019.

[14] Gers, Felix A., Jürgen A. Schmidhuber, and Fred A. Cummins. "Learning to Forget: Continual Prediction with LSTM." Neural Comput. 12, no. 10 (October 2000): 2451–2471. https://doi.org/10.1162/089976600300015015.

[15] Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." ArXiv:1412.3555 [Cs], December 11, 2014. http://arxiv.org/abs/1412.3555.

[16] Schuster, M., and K.K. Paliwal. “Bidirectional Recurrent Neural Networks.” IEEE Transactions on Signal Processing 45, no. 11 (November 1997): 2673–81. https://doi.org/10.1109/78.650093.

[17] DANIEL W. GRIFFIN and JAE S. LIM, Signal Estimation from Modified Short-Time Fourier Transform, IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. ASSP-32, NO. 2, APRIL 1984.