pISSN 2005-8063
eISSN 2586-5854
2019. 6. 30.
Vol.11 No.2
pp. 57-64

말소리와 음성과학

Phonetics and Speech Sciences

한국음성학회지

https://doi.org/10.13064/KSSS.2019.11.2.057

Performance comparison of various deep neural network architectures using Merlin toolkit for a Korean TTS system*

Junyoung Hong[1] · Chulhong Kwon[2]**

[1]*Selim TSG Co., Daejeon, Korea*
[2]*Department of Electronics, Information & Communication Engineering, Daejeon University, Daejeon, Korea*

Abstract

In this paper, we construct a Korean text-to-speech system using the Merlin toolkit which is an open source system for speech synthesis. In the text-to-speech system, the HMM-based statistical parametric speech synthesis method is widely used, but it is known that the quality of synthesized speech is degraded due to limitations of the acoustic modeling scheme that includes context factors. In this paper, we propose an acoustic modeling architecture that uses deep neural network technique, which shows excellent performance in various fields. Fully connected deep feedforward neural network (DNN), recurrent neural network (RNN), gated recurrent unit (GRU), long short-term memory (LSTM), bidirectional LSTM (BLSTM) are included in the architecture. Experimental results have shown that the performance is improved by including sequence modeling in the architecture, and the architecture with LSTM or BLSTM shows the best performance. It has been also found that inclusion of delta and delta-delta components in the acoustic feature parameters is advantageous for performance improvement.

Keywords: deep neural networks, Merlin toolkit, text-to-speech (TTS)

1. 서론

텍스트-음성 변환(text-to-speech, TTS) 시스템은 임의의 텍스트를 입력으로 음성신호를 합성한다. TTS 시스템에서 은닉 마르코프 모델(hidden Markov model, HMM) 기반의 통계적 파라미터 음성합성(statistical parametric speech synthesis, SPSS) 방식이 (Yoshimura et al., 1999) 널리 사용되고 있다. 이 방식은 음성 파형 연결 방식 또는 unit selection 방식(Hunt & Black, 1996)과 비교하여 기본 주파수(F0), 지속시간(duration), 음색 등을 조절할 수 있고 음성 DB가 적게 필요하다는 등의 이점을 갖고 있지만, 상대적으로 합성 음성의 품질이 떨어지는 단점을 갖고 있다(Zen et al., 2013).

HMM 기반의 SPSS 방식에서 합성 음성의 품질을 저하시키는 원인 중의 하나는 문맥 요인을 포함시키는 음향 모델링의 구성

* This work was supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program (No. 10080667), and supported by the Daejeon University Research Grants (20173595).
** chkwon@dju.ac.kr, Corresponding author
Received 31 January 2019; Revised 8 March 2019; Accepted 27 March 2019

에 있다고 알려져 있다(Ling et al., 2015; Zen et al., 2009). 음향 모델링은 입력 텍스트로부터 추출한 언어 특징과 보코더에 필요한 음향 특징 사이의 관계를 연결한다. HMM에서는 언어 및 음향적으로 관련된 문맥 요인을 회귀 트리를 사용하여 문맥에 의존적으로 클러스터링 한다. 회귀 트리 방식으로 문맥에 의존적인 모든 HMM 모델을 예측하기 위해서는 방대한 크기의 트리가 필요하게 되고 따라서 문맥 간 평균화를 하게 되는데 이는 합성음성의 품질을 저하시키는 요인으로 작용한다(Merritt et al., 2015; Yu et al., 2011). 그러므로 회귀 트리보다 성능이 우수한 모델을 사용하는 방법을 모색할 필요성이 제기된다.

심층 신경망 알고리즘은 고차원의 복잡한 추상화와 데이터 표현을 추출하는 능력을 보여주고 있으며(Bengio, 2009; Najafabadi et al., 2015), 음성 인식, 영상 처리, 기계 번역 등 다양한 분야에서 상당한 성능 개선을 보여주었다. 그리하여 심층 신경망이 SPSS 방식에서 언어 특징과 음향 특징 사이의 관계를 학습하기 위한 음향 모델링의 대안으로 부상하고 있다(Weijters & Thole, 1993; Zen et al., 2013). 또한, 음성합성에 필요한 지속시간을 예측하기 위한 모델링 방법으로 심층 신경망을 이용할 수 있다(Riedi, 1995). 본 논문에서는 음향 모델링을 구성하기 위하여 심층 신경망 기법을 적용하는 방안에 대하여 연구한다.

본 논문에서는 Merlin 툴킷을 소개하고 이를 이용하여 한국어 TTS 시스템을 구성하는 방법론을 제시한다. 그리고 심층 신경망 분야에서 널리 사용되는 다양한 심층 신경망 기법을 적용하여 음향 모델링 아키텍처를 제시하고, 이들의 성능을 비교 실험하여 우수한 성능을 보여주는 방법을 제안한다.

2. Merlin 툴킷

본 논문에서 사용한 Merlin 툴킷은 Edinburgh 대학의 음성 연구센터가 공개한 음성 합성을 위한 오픈소스 시스템이다(Wu et al., 2016). Merlin은 보코더에 필요한 음향 특징 파라미터의 추출, 신경망을 통한 음향 모델링 학습 기능과 음성파형 생성을 위한 보코더 기능을 포함하고 있다. Merlin은 파이썬 언어로 작성되어 있으며, 소스 코드 문서와 다양한 시스템 구성을 위한 레시피가 포함되어 있다.

Merlin은 자체적인 front-end 전처리 기능을 완벽하게 제공하고 있지 않기 때문에 기본적으로 Festival(CSTR, 2014)이나 Ossian (CSTR, 2018a)과 같은 외부의 추가적인 전처리 모듈이 필요하다. 하지만 다른 텍스트 전처리 모듈과의 인터페이스가 간단하다. 음성 DB와 이에 대응하는 텍스트 그리고 질문 셋을 이용하여 입력 텍스트를 Merlin에 적합한 HTS(Nitech, 2015) 스타일의 레이블 포맷으로 변환할 수 있도록 지원한다.

지속시간 모델과 음향 모델이 복잡한 비선형적인 특성을 가지므로 Merlin에서는 심층 신경망을 사용하여 이를 학습한다. 심층 신경망의 각 은닉 계층에 층마다 다른 종류의 신경망 구조를 사용할 수 있다. 이 구조에는 전연결 심층 피드포워드 신경망(fully connected deep feedforward neural network, DNN), 순환 신경망(recurrent neural network, RNN), 장단기 기억 신경망(long short-term memory, LSTM), 게이트 순환 신경망(gated recurrent unit, GRU), 양방향 LSTM(bidirectional LSTM, BLSTM) 등을 포함할 수 있다. 심층 신경망 학습에 필요한 다양한 활성화 함수, 파라미터의 다양한 초기화 방법 등의 기능을 제공하는데, 각 계층의 활성화 함수로는 선형 함수, 쌍곡선 탄젠트(hyperbolic tangent), logistic sigmoid, rectified linear unit(ReLU), rectified smooth unit (ReSU) 등을 지원한다. 신경망의 주요한 하이퍼-파라미터들 중 학습률(learning rate), dropout rate, 배치 사이즈, 학습 에포크의 수 등을 간단한 설정을 통해 변경할 수 있다(Wu et al., 2016).

언어 및 음향 특징 벡터 정규화, F0 보간, 음향 특징 벡터 구성 등의 기능도 제공한다. 학습을 통해 생성한 지속시간과 음향 특징 벡터로부터 음성신호를 만들기 위해 사용하는 보코더로는 WORLD(Morise et al., 2016)와 STRAIGHT(Kawahara et al., 1999) 등을 지원하며 필요에 따라 다른 보코더를 이용할 수 있다.

음성을 합성할 때 부드러운 음향 특징 파라미터 궤적을 생성하기 위해 maximum likelihood parameter generation(MLPG) 기법(Tokuda et al., 1995)이 적용되고, 명료도 증가를 위해 켑스트럼 영역에서 스펙트럼 개선 방법이 멜 켑스트럼 파라미터에 적용된다(Wu et al., 2016).

3. 음향 모델링에 사용하는 심층 신경망

언어 특징을 음향 특징에 매핑하기 위하여 DNN 방법을 모델로 사용할 수 있다. 이 모델은 여러 개의 은닉 계층으로 구성되어 있고 은닉 계층 간에는 전연결되어 있다. 입력 벡터는 은닉 계층의 출력을 예측하는 데 사용되고, 은닉 계층은 각각 이전 계층의 출력에 비선형 활성화 함수를 적용한다. 그러나 이 방식은 음성신호와 같은 시계열 데이터의 주요 특징인 장시간 문맥 정보를 포함하기가 어렵다고 알려져 있다(Williams & Zipser, 1992).

문맥 정보를 포함하여 언어 특징 시퀀스를 음향 특징 시퀀스로 매핑하도록 시퀀스 모델링 문제로 풀기 위해 RNN 방식을 이용할 수 있다. 이 방식은 입력 텍스트의 전체 히스토리를 이용하여 신경망을 업데이트 하는 구조로 구성되어 있다. 이 경우에 순환 연결은 음향 특징 시퀀스의 정보를 매핑하고 기억할 수 있고, 이러한 음향 특징 시퀀스는 음성신호 처리에서 출력의 예측을 높이기 위해 중요하다. 그러나 RNN 방식은 기울기 소실(vanishing gradient) 문제가 있어 장시간의 정보를 기억하는데 어려움이 있다고 알려져 있다(Bengio et al., 1994).

LSTM은 RNN 방식에서 학습 중에 발생하는 기울기 소실 문제를 해결하기 위한 구조를 가진 순환 신경망의 일종이며, 잠재적인 장시간 기억 의존성을 유지한다(Hochreiter & Schmidhuber, 1997). 따라서 시계열 신호를 분류, 처리 및 예측하기 위해 LSTM은 히스토리로부터 학습할 수 있다. LSTM은 시간이 지남에 따라 그 상태를 유지하기 위해 순환적 은닉 계층에 자체 연결을 갖는 특별한 메모리 셀과 이전 상태를 기억하며 각 계층의 입력과 출력에 정보의 흐름을 제어하는 데 사용되는 3개의 게이트 구조(입력 게이트, 망각 게이트 및 출력 게이트)를 갖는다. 이러

한 순환 출력 계층을 갖는 LSTM은 입력 텍스트의 문맥 정보를 포착한다고 생각할 수 있다.

BLSTM은 좌측에서 우측 방향으로의 순방향 상태 시퀀스와 우측에서 좌측 방향으로의 역방향 상태 시퀀스로 처리하는 2개의 LSTM 출력을 연결함으로써 작업을 수행한다(Schuster & Paliwal, 1997). 단방향 LSTM은 과거의 시간 인스턴스에서 온 문맥 정보만 고려되는 반면에, BLSTM은 순방향과 역방향에서 전달하는 과거와 미래의 문맥 정보를 모두 이용하여 학습할 수 있다.

LSTM을 약간 단순화한 신경망인 GRU 아키텍처가 제안되었다(Chung et al., 2014). LSTM은 3개의 게이트와 메모리 셀을 갖는 반면에, GRU는 업데이트 게이트와 리셋 게이트 등 2개의 게이트로 구성되어 있고 별도의 메모리 셀은 없다. GRU에서 업데이트 게이트는 LSTM의 망각 게이트와 같이 장시간 의존성을 포착한다. GRU에서는 출력 게이트가 사용되지 않기 때문에 GRU 파라미터의 수는 LSTM보다 적고 오버피팅을 피할 수 있다.

4. 실험 방법

4.1. 실험 환경

본 논문에서는 Merlin 툴킷을 이용하여 한국어 음성 데이터와 음성에 대응하는 텍스트 파일을 입력으로, 여성 전문 성우의 목소리를 기초로 한 한국어 TTS 시스템을 구성한다. 총 발화는 5,000개이고, 이 중에서 4,800개는 학습 데이터, 100개는 검증 데이터, 100개는 테스트 데이터로 사용하며, 녹음 분량은 약 8시간 40분이다. 잡음이 없는 조용한 스튜디오 환경에서 녹음하였고, 샘플링 주파수는 16 kHz, 선형 PCM 16비트 포맷으로 저장되었다.

Merlin 툴킷에서는 음성 신호를 생성하기 위한 보코더로 STRAIGHT와 WORLD를 지원하는데, STRAIGHT는 오픈 소스가 아니고 WORLD는 오픈 소스이므로 본 논문에서는 WORLD 보코더를 사용한다. 이 보코더에 필요한 특징 파라미터는 60차원의 Mel-cepstral coefficients(MCEP), 1개의 band aperiodicity(BAP) 및 1개의 로그 F0와 이들의 델타 및 델타-델타를 포함하고, 유성음/무성음 분류 특징 1개 등 프레임 당 총 187개의 파라미터로 구성되고, 이 파라미터를 음성 DB에서 5 msec 프레임 간격으로 추출한다(Morise et al., 2016). Merlin 툴킷에서는 BAP 파라미터의 개수를 STRAIGHT를 사용할 시 25개, WORLD를 사용할 시 샘플링 주파수에 따라 48 kHz에서는 5개, 16 kHz에서는 1개를 추출하도록 설정되어 있다.

Merlin 툴킷은 리눅스 환경에서 작동하므로 사용한 운영체제는 Ubuntu 16.04 LTS이며, GPU는 Nvidia GTX 1080ti이다. Merlin은 완전한 시스템이 아니므로, 본 논문에서는 tensorflow를 이용하여 4.2.절에서 기술한 신경망 모델을 구현하였다. 즉, RNN, GRU, LSTM, BLSTM 모델을 tensorflow의 최신 라이브러리를 이용하여 구현하였다. 최적화는 Adam 방식을 사용하고, 학습률은 0.002에서 시작하여 기하급수적인 감소(exponential decay) 방식을 적용하고, 배치 사이즈는 128 등을 사용하였다. 이러한 하이퍼-파라미터는 4.2.절의 6개 시스템에 동일한 조건으로 적용하였다.

4.2. 음향 모델링 아키텍처 구성

본 논문에서는 음향 모델링 아키텍처 구성 방법으로 다음과 같이 시스템 A에서 F까지 6가지를 제시한다. 입력 벡터는 언어 정보를 포함하는 HTS 스타일의 레이블이고, 출력 벡터는 WORLD 보코더에서 음성신호를 생성하는데 필요한 음향 특징 파라미터이다. 시스템 B에서 F까지 은닉 계층을 구성하는 5개의 신경망 중에서 앞의 4개 'TANH'는 음향 모델의 특징을 추출하는 과정으로, 마지막 1개는 전후 문맥을 고려하는 성분을 추출하는 과정으로 볼 수 있다. 6개의 시스템 구성에서 은닉층의 수, 은닉층 노드의 수, 활성화 함수 종류 등은 성능 비교를 위하여 가능한 한 동일하게 유지하였다.

4.2.1. 시스템 A

6개의 DNN 은닉 계층을 사용하며, 각 은닉 계층은 1,024개의 노드로 구성되어 있고 활성화 함수로는 쌍곡선 탄젠트(TANH) 함수를 사용한다. 이와 같은 구조를 다음과 같이 설정(configuration) 파일에 저장한다.

· 은닉 계층 신경망 모델 : ['TANH', 'TANH', 'TANH', 'TANH', 'TANH', 'TANH']

· 은닉 계층 크기 : [1024, 1024, 1024, 1024, 1024, 1024]

4.2.2. 시스템 B

이 시스템은 1,024개의 노드와 쌍곡선 탄젠트 함수를 활성화 함수로 사용하는 4개의 DNN 은닉 계층(시스템 A와 동일)과 512개의 노드를 가진 하나의 RNN 계층으로 구성된 하이브리드 아키텍처이다.

· 은닉 계층 신경망 모델 : ['TANH', 'TANH', 'TANH', 'TANH', 'RNN']

· 은닉 계층 크기 : [1024, 1024, 1024, 1024, 512]

4.2.3. 시스템 C

시스템 B와 같은 구조이나, RNN 대신에 GRU를 사용한다.

· 은닉 계층 신경망 모델 : ['TANH', 'TANH', 'TANH', 'TANH', 'GRU']

· 은닉 계층 크기 : [1024, 1024, 1024, 1024, 512]

4.2.4. 시스템 D

이 시스템도 아키텍처는 시스템 B와 같으나, RNN을 BLSTM으로 대체한다. BLSTM의 은닉 계층 크기가 다른 방식과 달리 256인 것은 내부적으로 순방향과 역방향 시퀀스에 각각 256을 할당하여 은닉 계층의 전체 크기를 512로 처리하기 때문이다.

· 은닉 계층 신경망 모델 : ['TANH', 'TANH', 'TANH', 'TANH', 'BLSTM']

· 은닉 계층 크기 : [1024, 1024, 1024, 1024, 256]

4.2.5. 시스템 E

아키텍처는 시스템 B와 같으나, RNN 대신에 LSTM을 사용한다.

· 은닉 계층 신경망 모델 : ['TANH', 'TANH', 'TANH', 'TANH', 'LSTM']

· 은닉 계층 크기 : [1024, 1024, 1024, 1024, 512]

4.2.6. 시스템 F

아키텍처는 시스템 E와 동일하다.

· 은닉 계층 신경망 모델 : ['TANH', 'TANH', 'TANH', 'TANH', 'LSTM']

· 은닉 계층 크기 : [1024, 1024, 1024, 1024, 512]

시스템 A부터 E까지는 음성 특징 파라미터의 델타 및 델타-델타 특징을 포함하여 총 187개의 파라미터를 사용하지만, 시스템 F는 이 파라미터들을 사용하지 않는다. 즉, MCEP 60개, BAP 1개, 로그 F0 1개, 유성음/무성음 분류 특징 1개 등 총 63개의 파라미터를 사용한다. 시스템 F는 시스템 E와 비교를 위해 구성하였고, 앞의 다섯 가지 시스템 중에서 명료도 측면에서 가장 우수한 성능을 보이는 시스템 E에만 이를 적용하여 비교 실험하였다.

4.3. 실험 절차

입력 데이터로부터 4.2.절에서 제시한 모델을 학습하고 학습된 모델을 이용하여 음성을 합성하는 절차(CSTR, 2018b)를 간략하게 설명한다. 총 7개의 과정으로 나누어 기술한다.

4.3.1. Setup.sh

기본적인 입력 데이터 폴더(database)와 음향 모델과 지속시간 모델을 생성할 폴더(experiment/acoustic_model, experiment/duration_model)와 학습된 모델을 이용하여 생성한 합성 음성을 저장하는 폴더(test_synthesis)를 만든다. Global_settings.cfg라는 참조 파일을 만들어 학습용 파일 개수(train=4,800), 검증용 파일 개수(valid=100), 테스트용 파일 개수(test=100), 학습용 음성 파일의 리스트(file_id_list.scp), 사용할 보코더 이름(WORLD), 샘플링 주파수(16 kHz) 등을 참조하는데 활용된다. 또한, Merlin을 구동하는데 필요한 음성신호처리 툴인 speech signal processing toolkit(Imai & Kobayashi, 2017)의 경로를 저장해 둔다.

4.3.2. Prepare_labels.sh

준비된 음성 파일(database/wav)과 이에 해당하는 텍스트 파일(database/txt) 그리고 질문 셋 파일을 이용하여 HTS 스타일의 레이블 파일(database/labels/label_state_align/ *,lab)을 만든다. Merlin에서 제공하는 작업은 영어에 대해 수행하므로, 본 연구에서는 이 과정을 수행하지 않고 별도의 과정을 거쳐 사전에 해당되는 파일을 만들어 사용하였다.

4.3.3. Prepare_acoustic_features.sh

Merlin에서 제공하는 WORLD 보코더를 이용하여 준비된 음성 파일(database/wav)로부터 MCEP, BAP, 로그 F0 등의 음향 특징 파라미터를 추출하여 폴더(database/feats)에 저장한다. 이 데이터를 음향 모델을 학습하는 데 사용한다.

4.3.4. Prepare_conf_files.sh

지속시간 모델과 음향 모델의 학습에 필요한 설정 파일인 duration.conf와 acoustic.conf와 합성 시에 필요한 test_dur_synth.conf와 test_synth.conf 파일을 만든다. 이러한 설정 파일에는 은닉 계층 신경망 모델, 은닉 계층 크기, 최적화 방법, 학습률, dropout_rate, 배치 사이즈, 학습 에포크의 수 등을 지정할 수 있다.

4.3.5. Train_duration_model.sh

지속시간 모델을 학습시키는 과정으로, 4.3.4.에서 생성된 duration.conf 파일을 참조하여 학습한다.

4.3.6. Train_acoustic_model.sh

음향 모델을 학습시키는 과정으로, 4.3.4.에서 생성된 acoustic.conf 파일을 참조하여 4.2.절에서 제시한 음향 모델링 아키텍처를 학습한다.

4.3.7. Run_merlin.sh

4.3.2.의 과정에서처럼 사전에 HTS 스타일로 작성된 레이블(test_synthesis/prompt_lab)과 합성할 텍스트 리스트(test_id_list.scp)를 입력으로 test_dur_synth.conf와 test_synth.conf 파일을 참조하여 4.3.6.과 4.3.7.에서 학습된 지속시간 모델과 음향 모델을 이용해 WORLD 보코더로 합성 음성을 생성한다.

5. 실험 결과

4.2.절에서 제시한 6개 시스템의 성능을 비교하기 위하여, 객관적인 성능 지표인 mel-cepstral distortion(MCD), band aperiodicity distortion(BAD), F0의 root mean squared error(RMSE)를 사용한다. 테스트 음성 파일 100개에 대해 원 음성과 합성 음성을 비교하여 값을 구하였다.

MCD는 원 음성과 합성 음성의 멜-켑스트럼 차이를 측정하는 지표로 단위는 [dB]이다(Kubichek, 1993; Luo et al., 2016).

$$\mathrm{MCD} = (10/\ln 10) * \frac{1}{N}\sum_{i=1}^{N}\sqrt{2\sum_{j=1}^{60}(mcep_{i,j}^{t} - mcep_{i,j}^{s})^2} \quad (1)$$

여기에서 $mcep^{t}, mcep^{s}$ 는 각각 원 음성과 합성 음성의 60차원의 MCEP이고, N은 전체 테스트 문장의 프레임 수이다.

BAD는 원 음성과 합성 음성의 BAP의 차이를 측정하는 지표로 단위는 [dB]이다.

$$BAD = (10/\ln 10) * \frac{1}{N}\sum_{i=1}^{N}\sqrt{2(bap_i^t - bap_i^s)^2} \quad (2)$$

여기에서 bap^t, bap^s 는 각각 원 음성과 합성 음성의 1차원의 BAP이다.

F0의 RMSE는 원 음성과 합성 음성의 F0 차이를 측정하는 지표로 단위는 [Hz]이다.

$$F0\_RMSE = \sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p}(F0_i^t - F0_i^s)^2} \quad (3)$$

여기에서 $F0^t, F0^s$ 는 각각 원 음성과 합성 음성의 $F0$이고, N_p 은 전체 테스트 문장의 유성음 구간의 프레임 수이다.

표 1에서 보이는 성능을 비교해 보면, 시퀀스 모델을 적용하지 않은 시스템 A의 성능이 가장 나쁘고, 시퀀스 모델을 적용하는 시스템 B, C, D, E가 더 우수하다는 결과를 알 수 있는데, 이로부터 문맥을 고려하는 시퀀스 모델을 아키텍처에 포함하는 것이 바람직하다는 것을 알 수 있다.

표 1. 아키텍처별 성능 비교
Table 1. Performance comparison for various architectures

	MCD [dB]	BAD [dB]	F0 RMSE [Hz]	합성시간 (상대 비)	모델 사이즈 (MB)
시스템 A	5.635	0.263	18.375	1	71.8
시스템 B	4.986	0.233	15.774	3.802	54.9
시스템 C	4.800	0.230	15.216	4.192	73.8
시스템 D	4.772	0.226	13.443	4.584	83.3
시스템 E	4.738	0.226	13.785	3.938	83.2
시스템 F	4.824	0.230	14.360	-	82.5

RNN이 적용된 시스템 B는 장시간 기억 의존 능력이 떨어져 시퀀스 모델 중에서 성능이 가장 나쁜 것으로 나타났다. GRU (시스템 C)는 LSTM(시스템 E)과 비교하여 구조가 간단해 학습 모델의 파라미터 수가 적어 적은 데이터로도 학습이 가능할 수 있지만, 충분한 수의 데이터가 있을 경우에는 LSTM 모델링이 더 좋은 결과를 보여줄 수 있다는 것을 알 수 있다. LSTM이 적용된 시스템 E가 BLSTM이 적용된 시스템 D보다 명료성을 나타내는 MCD 성능 지표에서 성능이 좋게 나온 결과는 명료도 측면에서 역방향 시퀀스의 중요도가 떨어진다는 것을 보여준다. 자연성을 나타내는 F0 예측 능력은 BLSTM이 적용된 시스템 D가 성능이 가장 우수하다.

시스템 E와 F를 비교해 보면, 시스템 E는 F와 달리 음향 특징의 델타와 델타-델타 파라미터를 사용하여 성능이 더 좋음을 볼 수 있는데, 이는 LSTM이 문맥 정보를 이용한다고 하더라도 여전히 음향 특징의 델타와 델타-델타 파라미터를 사용하는 것이 성능 개선에 유리하다는 것을 알 수 있다.

합성 시간은 시퀀스 모델을 적용한 경우(시스템 B, C, D, E)가 그렇지 않은 경우(시스템 A)보다 4배 정도 많이 걸리고, 그 중에서도 BLSTM을 적용한 시스템 D가 가장 오래 걸린다는 것을 알 수 있다. 시스템 F는 다른 시스템과는 달리 음향 특징의 델타와 델타-델타 파라미터를 사용하지 않으므로 합성 시간 비교 대상에서 제외하였다.

합성 음성을 청취 평가한 결과를 명료도와 자연성 측면에서 정성적으로 분석한 결과는 다음과 같다.

시퀀스 모델을 적용하지 않은 시스템 A는 이 모델을 적용한 시스템보다 합성음의 명료도가 떨어져 음가가 분명하게 들리지 않는다. 또한, 운율이 생생하지 않고 음조가 평탄하여 기계음적인 합성음으로 들려, 시퀀스 모델이 자연성 개선에 크게 기여하는 것으로 보인다.

RNN을 적용한 시스템 B는 시스템 C, D, E와 비교하여 명료도 측면에서는 유사한 성능을 보이나, 입력 텍스트가 긴 문장의 일부분에서 운율이 단조로워 소리의 생생함이 떨어지는 결과를 보여주는데, 이는 RNN이 장시간 기억 능력이 부족하다는 연구 결과에 기인한다고 볼 수 있다. 그 외 시스템 C, D, E는 청취 평가에서 미세한 차이를 보여 성능의 차이를 구분하기 어려웠다.

음향 특징의 델타와 델타-델타 파라미터를 사용하지 않은 시스템 F는 이를 사용한 시스템보다 공명이 더 심해 우는 소리가 들린다.

6. 결론

본 논문에서는 먼저 Edinburgh 대학의 음성 연구센터가 공개한 음성 합성을 위한 오픈소스 시스템인 Merlin 툴킷을 소개하였다. 그리고 여성 전문 성우의 목소리로 음성 DB를 구축하고 Merlin 툴킷을 이용하여 한국어 TTS 시스템을 구성하였다.

TTS 시스템에서 HMM 기반의 SPSS 방식이 널리 사용되고 있는데, 이 방식은 문맥 요인을 포함시키는 음향 모델링을 구성할 때 회귀 트리를 사용하여 음향 모델을 구성하고 문맥간 평균화를 하게 되는데, 이것으로 인해 합성음성의 품질이 저하된다고 알려져 있다. 따라서 회귀 트리보다 성능이 우수한 모델을 사용하는 방법이 필요하다.

본 논문에서는 여러 분야에서 우수한 성능을 보여 주는 심층 신경망 기법을 적용하여 음향 모델링 아키텍처를 제안하였다. 이 구조에는 DNN, RNN, GRU, LSTM, BLSTM 등이 포함되어 있다. Merlin은 완전한 시스템이 아니므로 이러한 심층 신경망 모델을 tensorflow의 최신 라이브러리를 이용하여 구현하였다.

음향 모델링 아키텍처 구성 방법으로 시스템 A에서 F까지 6가지를 제시하였는데, 각 아키텍처에서 앞의 신경망 모델은 음향 모델의 특징을, 마지막 신경망 모델은 문맥을 고려하는 성분을 추출하는 과정으로 볼 수 있다. 실험 결과, 문맥을 고려하는 시퀀스 모델을 아키텍처에 포함하는 것이 성능 개선에 유리하다는 것을 알 수 있고, 명료도 측면에서는 LSTM을 적용한 아키텍처가, 자연성 측면에서는 BLSTM을 적용한 아키텍처가 가장 좋은 성능을 보여주었다. 그리고 음향 특징 파라미터에 델타와 델타-델타 성분을 포함하는 것이 성능 개선에 유리하다는 결과가 도출되었다.

향후 연구로, 심층 신경망 기법은 매우 많은 데이터를 필요로 하므로 음성 DB를 추가로 수집하여 합성 음성의 품질 개선에 대해 실험할 계획이다. 그리고 TTS뿐만 아니라 음색 변환(voice conversion)이나 화자 적응 분야에도 Merlin 툴킷을 이용하여 연구를 진행할 계획이다.

References

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157-166.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, *2*(1), 1-127.

Chung, J., Gulcehre, C., Cho, K. H., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. Retrieved from https://arxiv.org/abs/1412.3555

CSTR [The Center for Speech Technology Research]. (2014). Festival: The festival speech synthesis system (version 2.4) [Computer program]. Retrieved from http://www.cstr.ed.ac.uk/projects/festival/

CSTR [The Center for Speech Technology Research]. (2018a). Ossian: A python based tool for automatically building speech synthesis front ends [Computer program]. Retrieved from https://github.com/CSTR-Edinburgh/Ossian/

CSTR [The Center for Speech Technology Researc]. (2018b). The Merlin toolkit [Computer program]. Retrieved from https://github.com/CSTR-Edinburgh/merlin/tree/master/egs/build_your_own_voice/

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.

Hunt, A. J., & Black, A. W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. *Proceedings of the International Conference on Acoustics*, *Speech*, *Signal Processing* (pp. 373-376).

Imai, S., & Kobayashi, T. (2017). SPTK: Speech signal processing toolkit (version 3.11) [Computer program]. Retrieved from http://sp-tk.sourceforge.net/

Kawahara, H., Masuda-Katsuse, I., & de Cheveigne, A. (1999). Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication*, *27*(3-4), 187-207.

Kubichek, R. (1993). Mel-cepstral distance measure for objective speech quality assessment. *Proceedings of the IEEE Pacific Rim Conference on Communications Computers and Signal Processing* (pp. 125-128). Victoria, BC, Canada.

Ling, Z. H., Kang, S. Y., Zen, H., Senior, A., Schuster, M., Qian, X. J., Meng, H. M., & Deng, L. (2015). Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques & future trends. *IEEE Signal Processing Magazine*, *32*(3), 35-52.

Luo, Z., Takiguchi, T., & Ariki, Y. (2016). Emotional voice conversion using deep neural networks with MCC and F0 features. *Proceedings of the IEEE/ACIS 15th International Conference on Computer and Information Science* (pp. 1-5). Okayama, Japan.

Merritt, T., Latorre, J., & King, S. (2015). Attributing modelling errors in HMM synthesis by stepping gradually from natural to modelled speech. *Proceedings of the International Conference on Acoustics*, *Speech*, *Signal Processing* (pp. 4220-4224). Brisbane, Australia.

Morise, M., Yokomori, F., & Ozawa, K. (2016). WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, *E99.D*(7), 1877-1884.

Najafabadi, M., Villanustre, F., Khoshgoftaar, T., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*(1), 1-21.

Nitech [Nagoya Institute of Technology]. (2015). HTS: HMM/DNN-based speech synthesis system (version 2.3) [Computer program]. Retrieved from http://hts.sp.nitech.ac.jp/

Riedi, M. (1995). A neural-network-based model of segmental duration for speech synthesis. *Proceedings of the Eurospeech 1995* (pp. 599-602).

Schuster, M., & Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673-2681.

Tokuda, K., Kobayashi, T., & Imai, S. (1995). Speech parameter generation from HMM using dynamic features. *Proceedings of the 1995 International Conference on Acoustics*, *Speech*, *Signal Processing* (pp. 660-663). Detroit, MI.

Weijters, T., & Thole, J. (1993). Speech synthesis with artificial neural networks. *Proceedings of the International Conference on Neural Networks* (pp. 1764-1769). San Diego, CA.

Williams, R. J., & Zipser, D. (1992). Gradient-based learning algorithms for recurrent networks and their computational complexity. In Y. Chauvin, & D. E. Rumelhart (Eds.), *Back-propagation: Theory*, *architectures and applications* (pp. 433-486). Hillsdale, NJ: Lawrence Erlbaum Associates.

Wu, Z., Watts, O., & King, S. (2016). Merlin: An open source neural network speech synthesis system. *Proceedings of the 9th ISCA Speech Synthesis Workshop* (pp. 202-207).

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., & Kitamura, T. (1999). Simultaneous modeling of spectrum, pitch and duration in HMM based speech synthesis. *Proceedings of the Eurospeech 1999* (pp. 2347-2350).

Yu, K., Zen, H., Mairesse, F., & Young, S. (2011). Context adaptive training with factorized decision trees for HMM-based statistical

parametric speech synthesis. *Speech Communication*, *53*(6), 914-923.

Zen, H., Tokuda, K., & Black, A. W. (2009). Statistical parametric speech synthesis. *Speech Communication*, *51*(11), 1039-1064.

Zen, H., Senior, A., & Schuster, M. (2013). Statistical parametric speech synthesis using deep neural networks. *Proceedings of the 2013 IEEE International Conference on Acoustics*, *Speech*, *Signal Processing* (pp. 7962-7966). Vancouver, BC.

• **홍준영 (Hong, Junyoung)**
세림티에스지(주) 연구원
대전시 유성구 테크노1로 62-16
Tel: 042-488-7900
Email: hjyk21@naver.com
관심 분야: 음성합성, 딥러닝

• **권철홍 (Kwon, Chulhong)** 교신저자
대전대학교 전자·정보통신공학과 교수
대전시 동구 대학로 62
Tel: 042-280-2555
Email: chkwon@dju.ac.kr
관심 분야: 음성합성, 딥러닝

Merlin 툴킷을 이용한 한국어 TTS 시스템의 심층 신경망 구조 성능 비교*

홍 준 영[1] · 권 철 홍[2]

[1]세림티에스지(주), [2]대전대학교 전자 · 정보통신공학과

국문초록

본 논문에서는 음성 합성을 위한 오픈소스 시스템인 Merlin 툴킷을 이용하여 한국어 TTS 시스템을 구성한다. TTS 시스템에서 HMM 기반의 통계적 음성 합성 방식이 널리 사용되고 있는데, 이 방식에서 문맥 요인을 포함시키는 음향 모델링 구성의 한계로 합성 음성의 품질이 저하된다고 알려져 있다. 본 논문에서는 여러 분야에서 우수한 성능을 보여 주는 심층 신경망 기법을 적용하는 음향 모델링 아키텍처를 제안한다. 이 구조에는 전연결 심층 피드포워드 신경망, 순환 신경망, 게이트 순환 신경망, 단방향 장단기 기억 신경망, 양방향 장단기 기억 신경망 등이 포함되어 있다. 실험 결과, 문맥을 고려하는 시퀀스 모델을 아키텍처에 포함하는 것이 성능 개선에 유리하다는 것을 알 수 있고, 장단기 기억 신경망을 적용한 아키텍처가 가장 좋은 성능을 보여주었다. 그리고 음향 특징 파라미터에 델타와 델타-델타 성분을 포함하는 것이 성능 개선에 유리하다는 결과가 도출되었다.

핵심어: 심층 신경망, Merlin 툴킷, text-to-speech(TTS)

* 이 논문은 산업통상자원부의 산업기술혁신사업의 지원(No.10080667) 및 2017학년도 대전대학교 교내학술연구비의 지원(20173595)에 의한 연구결과로 수행되었음.