한국어 FastSpeech2의 하이퍼파라미터 조절을 통한 최적화 연구

임수환, 박동건, 김홍국
광주과학기술원
{suwhoanlim, dongkeon, hongkook}@gist.ac.kr

A Study on the Optimization of Korean FastSpeech2 by Manipulating Model Hyperparameters

Suwhoan Lim, Dong Keon Park, Hong Kook Kim
Gwangju Institute of Science and Technology(GIST)

요 약

본 논문은 트랜스포머 구조를 채택한 FastSpeech2에 Korean Single speaker Speech Dataset (KSS) 데이터셋을 이용한 한국어 음성합성시스템의 최적화에 대해 다룬다. 병렬화가 불가능하여 학습 속도가 느리다는 내재적인 단점을 가진 recurrent neural network를 이용한 기존 음성합성시스템과는 달리 FastSpeech2는 트랜스포머 구조를 채택하였기에 병렬화가 가능하여 학습 속도가 빠를 뿐만 아니라, variance adaptor를 모델에 포함해 음성합성시스템의 고질적인 문제였던 일 대 다수 문제에 대한 해법을 제시하였다. 본 논문에서는 FastSpeech2의 일부 하이퍼파라미터를 조정한다면 이러한 FastSpeech2의 장점을 살릴 수 있을 뿐만 아니라 한국어 화자 음성 정보를 활용한 한국어 음성합성시스템 성능 향상의 가설을 바탕으로 실험을 진행하였다. 독립변수는 인코더 블록의 개수와 각 블록 내부에 존재하는 멀티헤드의 개수이며, 각 실험의 결과는 훈련을 통해 나온 손실함수 그래프와 28명의 피실험자를 대상으로 한 평균 의견 점수를 통해 정량적, 정성적으로 분석하였다.

Ⅰ. 서 론

음성합성시스템(Text-to-Speech, TTS)이란 어떠한 문자(grapheme) 혹은 음소(phoneme)를 입력으로 받아 spectrogram 혹은 waveform을 생성해내는 구조나 모델을 통칭한다. 고전적인 음성합성시스템에는 미리 녹음된 어절, 또는 음절을 이어붙이는 방법이 존재한다[1]. 이러한 방법은 합성하고자 하는 문장의 구성 요소가 데이터베이스에 존재한다면 쉽게 음성을 합성해 낼 수 있다는 장점이 있지만, 합성된 결과물은 결국 초기에 녹음된 음성이 가지는 특색을 가질 수밖에 없다는 단점이 존재하였다.

이러한 단점을 탈피하고자 전통적인 음성합성시스템에서 벗어나 인공신경망을 활용한 최초의 sequence-to-sequence 음성합성시스템이 바로 Tacotron이다[2]. Recurrent neural network (RNN)를 비롯한 gated recurrent unit (GRU)을 활용하여 기계학습을 진행한 Tacotron은 고전적인 이어붙이기(concatenation) 방법보다 더 사람에 가까운 음성을 합성해 낼 수 있었지만, 병렬화가 불가능하다는 RNN의 내재적인 한계로 인한 오랜 학습 시간이 그 단점으로 꼽혔다.

이러한 단점을 극복하고자 다양한 개선 시도가 이뤄졌고, 신경망을 활용한 음성합성시스템 중 주목받고 있는 시스템으로는 FastSpeech2가 있다[3]. FastSpeech2의 장점을 유지함과 동시에 한국어 데이터셋에 대해 적용하기 위하여 본 논문에서는 두 가지 하이퍼파라미터를 지정하였다: 1) 인코더 블록의 개수와 2) 인코더와 디코더 블록 내부에 존재하는 멀티헤드의 개수. 즉, 이들이 트랜스포머 블록에 직접적으로 끼치는 영향을 비교한다.

Ⅱ. 본론

기계학습 및 인공 신경망을 이용하여 음성합성시스템을 구출할 때의 고질적인 문제 중 하나는 일 대 다수 매핑 문제이다. 일 대 다수 매핑 문제란 음성은 음색, 톤, 크기 등 다양한 요소가 결합하여 있는데, 음성을 합성하고자 할 때 정확히 어떤 음성을 만들어야 하는지를 입력된 문장만을 가지고 판단하기에는 역부족이라는 상황을 의미한다. 이는 결과적으로 부정확한 음성 합성 결과물을 생성해 내게 된다는 것이다. 이러한 문제에 대해 FastSpeech2는 그림 1(a)와 같은 트랜스포머 구조[4]를 채택하여 빠른 학습이 가능하고, 또한 그림 1(b)와 같은 variance adaptor 모듈을 모델 아키텍처에 포함하여 학습 과정에 추가적인 정보를 제공하였다[3]. 또한 Montreal forced aligner (MFA)[5]를 통해 학습에 이용된 음성의 ground truth 값을 추출하였기에 잘못된 학습 데이터로 인한 오류 전파를 최소화하였다는 특징이 있다.

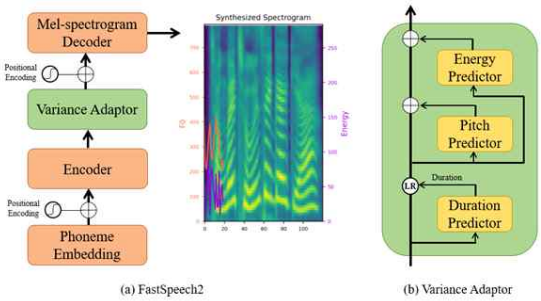

그림 1. (a) FastSpeech2와 (b) variance adaptor 구성도.

본 논문에서는 FastSpeech2의 훈련을 통해 나온 손실함수 값과 평균 의견 점수(Mean Opinion Score, MOS)를 통해 학습 결과를 판단한다. 이때 손실 함수는 정답 값과 mel-spectrogram의 평균절대오차와 variance adaptor의 세 가지 요소인 energy, pitch, 그리고 duration의 평균제곱오차의 합으로 되어있다.

Ⅲ. 실험

본 실험은 Korean Single speaker Speech (KSS) Dataset[6]을 이용하여 학습을 진행하였고, 그중 약 7.8%를 validation 데이터셋으로 사용하였다. FastSpeech2는 KSS Dataset에 맞춰진 Pytorch 구현을 사용하였으며

표 1. 음성 합성에 사용된 문장.

순번	문장
1	용돈을 아껴 써라.
2	그는 괜찮은 척하려고 애쓰는 것 같았다.
3	요즘 공부가 안돼요.

표 2. 인코더 레이어의 개수(E)에 따른 손실 함수 값 비교.

step=19,000	E1	E2	E4	E8
손실 함수	0.4603	0.4218	0.3995	**0.3827**

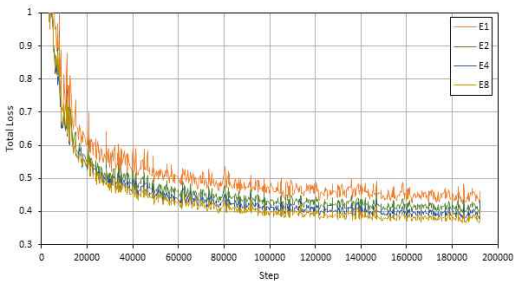

그림 2. 인코더 블록 개수(E)에 따른 손실 함수(total loss) 그래프.

표 3. 인코더 레이어의 개수(E)에 따른 95% 신뢰구간으로 나타낸 MOS 값 비교.

순번	GT	E=1	E=2	E=4	E=8
1	3.60±0.38	2.31±0.35	2.12±0.40	2.84±0.35	2.78±0.35
2	3.93±0.30	2.21±0.28	2.31±0.34	3.62±0.34	3.06±0.32
3	4.39±0.24	2.87±0.31	4.20±0.22	4.4±0.24	4.05±0.28
평균	3.82±0.41	2.38±0.35	2.77±0.49	3.47±0.44	3.17±0.41

[7][8], 정량적인 결과 분석에는 학습이 완료된 이후 손실 함수 그래프를 비교하였다. 또한 정성적인 분석을 위해서는 28명의 피실험자를 대상으로 평균 의견 점수(MOS)를 설문 받아 그 결과를 95%의 신뢰구간으로 표시하였다. KSS Dataset에서 MOS를 위해 사용한 문장은 표 1과 같다.

3.1 인코더 블록 변화 실험

첫 번째 실험에서 독립변수로 하는 하이퍼파라미터는 인코더 블록의 개수이다. 표 2와 그림 2에서 E는 인코더 블록 하이퍼파라미터를 뜻한다. 트랜스포머 블록의 은닉 차원은 256, 디코더 블록의 개수는 4, 멀티헤드의 개수는 2로 고정하였으며, 인코더 블록의 개수는 1, 2, 4, 8, 16으로 변경해 가며 결과를 측정하였다.

표 2에서 보는 바와 같이, 최종 스텝인 190,000에서 인코더 블록이 1개일 때와 8개일 때 손실 함숫값이 16.9% 차이 나는 것을 확인할 수 있다. 본 결과를 정성적으로 검증하기 위하여 28명의 피실험자를 대상으로 음질평가(MOS)를 진행한 결과는 표 3과 같다. 합성된 음성에서 vocoder의 불필요한 변수를 제거하기 위해 ground truth는 mel-spectrogram으로 변환된 이후 VocGAN[9]을 이용하여 음성으로 다시 합성하였다. 표 3에서는 인코더 블록의 개수가 증가함에 따라 MOS의 값 역시 증가하는 경향을 볼 수 있다. 다만 인코더 블록의 개수가 8개 일 때는 점수가 떨어졌는데, 이는 디코더의 개수가 고정된 상태에서 인코더 개수가 증가함에 따라 디코더의 개수를 초과하여 나타난 오버피팅으로 판단된다.

3.2 멀티헤드 개수 변화 실험

두 번째 실험에서 독립변수로 하는 하이퍼파라미터는 멀티헤드의 개수이다. 그림 3과 표 4는 트랜스포머 블록의 은닉 차원은 256, 인코더와 디코더 블록의 개수를 4로 고정하고 멀티헤드의 개수를 1, 2, 4, 8, 16개로 변경한 실험의 결과이다. 여기서, M은 멀티헤드 개수 하이퍼파라미터를 뜻한다.

표 4. 멀티 헤드 개수(M)에 따른 손실 함수 비교.

step=19,000	M1	M2	M4	M8	M16
손실 함수	0.4043	0.3995	0.3968	0.3925	**0.3904**

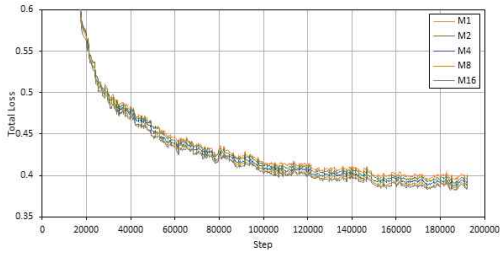

그림 3. 멀티헤드의 개수(M)에 따른 손실 함수(total loss) 그래프.

표 4에서는 멀티헤드 손실 함수 그래프에서 1개의 멀티헤드 개수일 때와 16개의 멀티헤드 개수일 때 손실 함숫값은 약 3.4% 차이를 보임을 확인할 수 있다. 이 실험을 통해 멀티헤드의 개수에 따라 손실 함숫값에 변동이 생김을 확인할 수는 있었지만, 인코더 블록 개수 실험보다 그 변화량이 적음을 볼 수 있었다. 이러한 차이는 트랜스포머 인코더 블록의 개수가 멀티헤드의 개수보다 훈련에 있어 더 주요한 역할을 수행하기 때문으로 판단된다. 실제로 멀티헤드 실험을 통해 합성된 한국어 음성을 정성적으로 분석해 보았을 때 음질평가를 진행할 정도로 유의미할 차이가 존재하지 않았다.

Ⅳ. 결론

본 논문에서는 FastSpeech2를 이용하여 한국어 데이터셋에 대해 최적화를 진행하였다. 본 실험은 FastSpeech2를 구성하는 요소 중 인코더 블록의 개수와 멀티헤드의 개수를 늘린다면 더욱 효율적인 기계학습이 가능하여 음성 합성 시스템의 성능 향상 가설을 수립하였다. 이는 실제로 실험 결과에서 손실 함숫값이 하락함과 동시에 음질평가를 통해 가설과 실험 결과가 일치함을 보였다. 다만 인코더 블록의 개수와 디코더 블록의 개수 등에 대한 퍼포먼스의 연관성을 입증하기 위해서는 추가적인 연구가 필요하다고 판단된다.

ACKNOWLEDGMENT

본 연구는 광주과학기술원 전기전자컴퓨터공학부 오디오지능연구실 인턴십의 결과이며, 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2019-0-01842, 인공지능대학원지원(광주과학기술원)).

참 고 문 헌

[1] H. Zen, K. Tokuda, and A. W. Black, “Statistical parametric speech synthesis,” *Speech Communication,* Vol. 51, No. 11, pp. 1039-1064, 2009.

[2] Y. Wang, et al.,“Tacotron: Towards End-to End Speech Synthesis,” *arXiv preprint arXiv:1703.10135,* 2017.

[3] Y. Ren, et al., “FastSpeech 2: Fast and high-quality end-to-end text to speech,” *arXiv preprint arXiv:2006.04558,* 2020.

[4] A. Vaswani, et al., “Attention is all you need,” *arXiv preprint arXiv:1706.03762,* 2017.

[5] M. McAuliffe, et al., “Montreal forced aligner: Trainable text-speech alignment using Kaldi,” in *Proc. Interspeech,* 2017.

[6] K. Park, *Korean Single Speaker Speech Dataset*, 2020, (https://www.kaggle.com/bryanpark/korean-single-speaker-speech-dataset)

[7] J. Kang, *Korean-FastSpeech2-Pytorch*, 2020, (https://github.com/HGU-DLLAB/Korean-FastSpeech2-Pytorch)

[8] S. Lim, *Korean-FastSpeech2-Pytorch*, 2021, (https://github.com/suwhoanlim/Korean-FastSpeech2-Pytorch)

[9] J. Yang, et al., “VocGAN: A high-fidelity real-time vocoder with a hierarchically-nested adversarial Network,” *arXiv preprint arXiv:2007.15256,* 2020.