Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 45 No. 7
July 2024

Efficient Deep Learning-Base Speech Synthesis Model Study for TTS

Seung-Yeon Hwang[1], Jeong-Joon Kim[2,*]
[1]Dept. of Computer Engineering, Anyang University, Anyang-si, Gyeonggi-do, Republic of Korea
[2,*]Corresponding Author, Dept. of Software, Anyang University, Anyang-si, Gyeonggi-do, Republic of Korea

Abstract: TTS technology using deep learning is very similar to human speech. In order to make such TTS technology, it receives text, creates a spectrogram is Text-to-Mel, and turns the spectrogram is Text-to-Mel, and turns the spectrogram into speech in a vocoder. However, there are various Text-to-Mel and Vocoder models. We need to devise a way to analyze and combine the various Text-to-Mel and Vocoders produce good sound quality. Therefore, in this paper, in order to convent English text to speech, a 24_hour LJ Speech Dataset that one person reads 7 books was used. And this dataset was used to train Text-to-Mel Models and vocoder models. Each learned Text-to-Mel and vocoder models were combined with each other, and the combination performance was compared for which one was the best combination and which one was the worst.
Keywords: End-to-End, Text-to-Mel, TTS, Vocoder

1. Introduction

Recently, as deep learning technologies have advanced, Text-to-Speech (TTS) technologeis have also begun to be influenced by deep learning. Traditional TTS systems produced poor audio quality and sounded robotic, but modern systems use deep learning to train on specific voices, creating outputs that are remakably similar to human speech. This technological advancement has expanded the use of TTS technologies in various fields, including, English language education [1]. The process of TTS involves receiving text input, generating a spectrogram via a Text-to-Mel process, and then synthesizing speech using a Vocoder.

There is a need to analyze and combine various Text-to-Mel and Vocoder models to produce the best sound quality. Therfore, this paper uses th LJ Speech Dataset, which contains 24 hous of audio from an individual reading seven books, to train different Text-to-Mel and Vocoder models. The study evaluates combinations of these trained models to determine which produce the best and worst synthetic speech.

The paper is organized as follows: Chapter 2 discusses realted work, Chapter 3 describes the experimental methods and results, and Chapter 4 concludes with the expected impacts of this research.

2. Related Research

1. Text-to-Speech (TTS)

Text-to-Speec h (TTS) technology converts text into speech. Traditionally, without the use of deep learning, creating synthetic speech involved complex processes that required cutting and combining recorded v oice data, necessitating specialized knowledge. Moreover, the quality often degraded due to the segmented development porcess, resulting in a roboic tone. Howe ver, with recent advances in deep learning, it is now po ssible to synthesize speech without extensive expertise in voice processing. Furthermore, end-to-end learning allows for training directly from text and speech data, producing high-quality voice synthesis that is robust ag ainst noise[2].

2. Text-to-Mel

The role of Text-to-Mel is to take text input and generate a spectrogram, which is then utilized by a Vovoder. This paper will employ the Tacotron2 and Fastspeech2 models.

2.1 Tacotorn2

DeDeveloped by Google, Tacotron2 emplys an autoreg ressive method to sequentially generate voice samples , which can be time-consuming and thus not suitable f or real-time processing environments. However, the ex tended synthesis time contributes to the high quality o f the synthetic speech. It allows for end-to-end learnin g with <text, audio> pairs and offers easy adjustments for features such as speaker, language, and emotion[2] . Tacotron2 consists of an Encoder, Attention, and Dec oder modules. The process flows from the Encoder to t he Attention module and then to the Decoder. Figure 1 below illustrates the Encoder, where an input sequenc e of integers transformed into one-hot vectors is first c onverted into a 512-dimensional embedding vector thr ough an Embedding Matrix. This transformed embeddi ng vector then passes through three convolutional laye rs and enters a bi-LSTM layer, resulting in encoded feat ures.

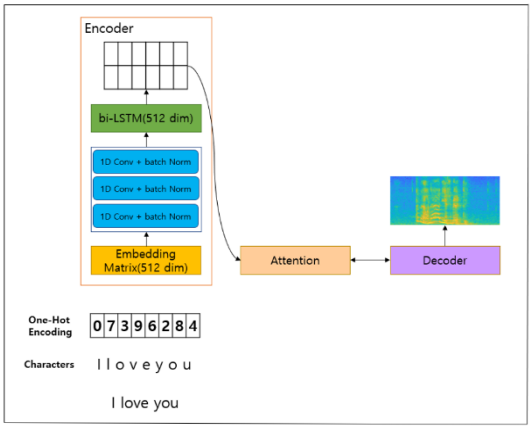

Fig 1. Tacotron2 Encoder Detailed Structure

Figure 2 below is about Attention. Attention serves to align w hich information to retrieve from the Encoder by utilizing the f eature generated from the LSTM of the Encoder and the featu re generated from the LSTM of the Decoder in the previous ti me step.

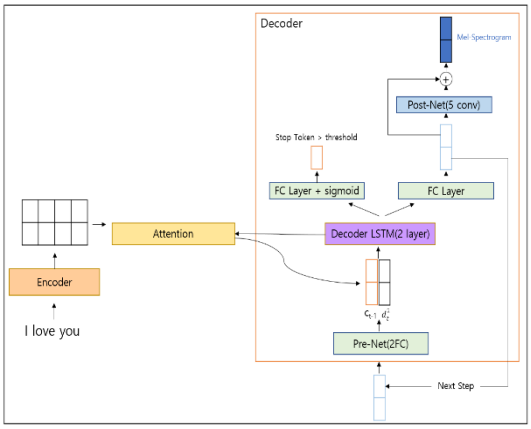

Fig 2. Tacotron2 Attention Detailed Structure

Finally, the Decoder uses the alignment feature obtain ed through Attention and the previously generated me l-spectrogram information to generate the next mel-sp ectrogram. The Decoder consists of a Pre-Net, Decoder LSTM, and Post-Net. The Pre-Net is composed of two F ully Connected Layers + ReLU. When the previously ge nerated mel-spectrogram enters as input to the decod er, it first passes through the Pre-Net. The Pre-Net acts as a bottleneck stage, filtering important information. The Decoder LSTM uses the information from the Atte ntion Layer and the information generated from the Pr e-Net to produce information specific to a certain time point. Lastly, the Post-Net consists of five 1D convoluti on layers and serves to enhance the quality of the final product, the mel-spectrogram. When the input enters the Decoder, it functions to extract and bring informati on from the Encoder that will be used at each time poi nt in the Decoder. When text is received, it converts it i nto numbers that represent it well. Next, the decoder i mplemented with RNN generates the next mel-spectro gram from the previous mel-scale spectrogram.

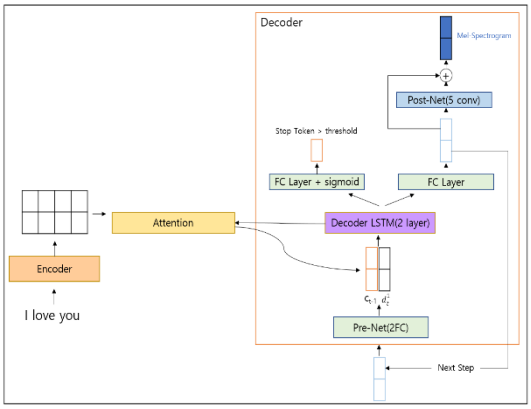

Fig 3. Tacotron2 Decoder Detailed Structure

2.2 FastSpeech2

FastSpeech2, unlike Tacotron2, employs a non-autoreg ressive method for generating speech samples in paral lel. As a result, the speech synthesis process is faster, e nabling real-time processing. However, the teacher-stu dent pipeline is complex and time-consuming, with dur ations extracted by the teacher model often not being sufficiently accurate, and the target mel-spectrograms extracted from the teacher model experiencing inform ation loss. Figure 3 below shows the architecture of Fa stSpeech2. As illustrated in Figure 2, the Variance Adap tor uses various variance data to accurately predict spe ech, properly fitting the model to prevent underfitting or overfitting. The features include a Duration Predicto r, Pitch Predictor, and Energy Predictor. The Duration P redictor determines how long each syllable prolongs th e speech, effectively turning phoneme duration into a key factor for predicting accuracy. It assesses how man y mel frames match each syllable and converts this inf ormation into a log to facilitate ease of prediction. The Pitch Predictor provides pitch, one of the elements tha t contribute to emotion, thus helping to produce more natural-sounding speech. The Energy Predictor measur es the magnitude of the mel-spectrogram and speech v olume, serving as one of the analytical elements of spe ech

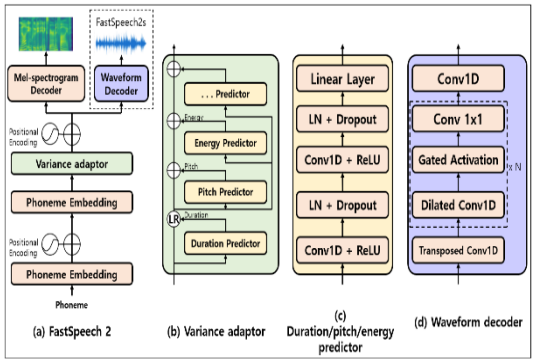

Fig 4. FastSpeech2 Architecture

3. Vocoder

The vocoder is the final stage in speech synthesis. It refers to a model that converts the Spectrogram produced by the Text-to-Mel model into an actual speech waveform. In this paper, we will use the WaveGlow, MelGAN, and Multi-band MelGAN models.

3.1 Wave Glow

WaveGlow is a vocoder model based on Normalizing Flow. As a flow-based vocoder, it enhances speed by parallelizing segments according to the group size relative to the sample rate across the entire time frame, and by generating samples for each group size within each segment, improving upon WaveNet. WaveGlow's process uses an invertible transformation function trained to produce simple distributions like Gaussian from a speech dataset. After training, the inverse function of the transformation is used to synthesize speech from samples of the Gaussian distribution. The quality and speed of the speech are acceptable, but it requires many parameters.

3.2 MelGAN

MelGAN is a vocoder model based on CNN and the Generative Adversarial Network (GAN). MelGAN's advantage is that it can be sufficiently trained with fewer parameters and produces stable sound quality relative to time. Figure 5 shows the architecture of the Generator in the MelGAN model. The process involves the Generator receiving the spectrum as input and consists of several layers of conv1d and a residual structure. Within the residual stack, there are additional conv1d layers, and it undergoes three residual connections.

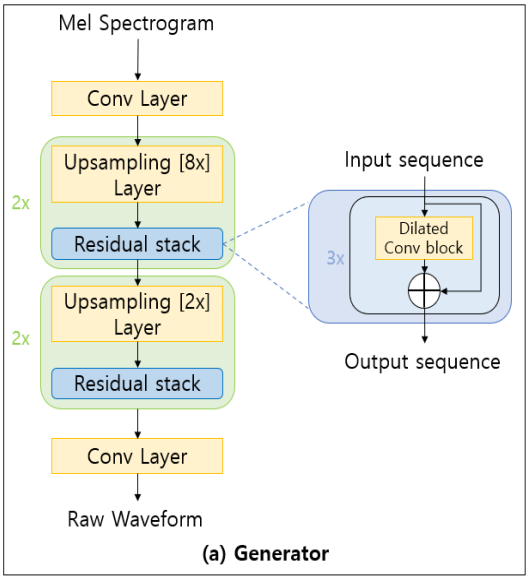

Fig 5. MelGAN Model Architecture(Generator)

Below, Figure 6 shows the architecture of the Discriminator in the MelGAN model. The Discriminator consists of three multi-scale structures. Each multi-scale has six feature maps and a total of seven outputs. Training without the multi-scale leads to issues with loss not converging. However, balancing the training of the discriminator is challenging because it is extremely sensitive to data, parameters, and model structures during training .

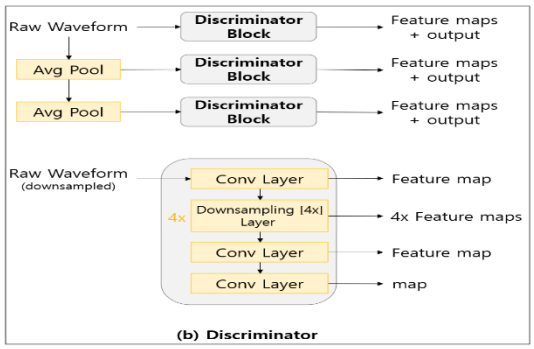

Fig 6. MelGAN Model Architecture(Discriminator)

2.3 Multi-band MelGANMulti-band MelGAN is an advanced model derived from the original MelGAN. While the original MelGAN helps stabilize the entire network by matching features, it is difficult to measure the latent features of true and fake voices. However, Multi-band MelGAN applies multi-resolution STFT loss, making it more effective at measuring the differences between true and fake voices.

2.4

3. Experimental Methods and Results

1. Experimental

EnvironmentTo implement this system, Python was used, and PyCharm served as the IDE. A GTX 1060 3GB GPU was used for training the models, and CUDA 10.1 was utilized when working with PyTorch. The English LJ Speech Dataset was used for training. This dataset features 13,100 short audio clips of a single person reading from seven books, all in the public domain. Transcriptions are provided for each clip. The clips vary in length from 1 to 10 seconds, totaling about 24 hours of data.

2. Experimental Process

To facilitate smooth synthesis, all Samplate values were standardized to 22050 before training. Considering the GPU environment, fp16_run was set to false, and the batch size was also set to a low value to ensure that training proceeded without significant issues.

3. Experimental Results

To measure the quality of the synthesized speech, a subjective evaluation was conducted with 30 participants using the MOS method. Ten sentences were synthesized, including interrogative sentences, declarative sent

ences, exclamations, and long sentences. The MOS scores range from 1 to 5, with higher scores indicating better synthesized sound quality.

3.1 Results with Tacotron2 + Vocoder Combination

The combination with MelGAN did not yield good sound quality but had proper intonation. The combination with Multi-band MelGAN resulted in good intonation and sound quality, though there were occasional issues with poor endings. The combination with WaveGlow was fast, and there were issues with poor endings, but the sound quality was good. Below, Table 1 shows the evaluation of synthesized sound when combining Tacotron2 and Vocoder. As seen in Table 1, Tacotron2 + Multi-band MelGAN received higher scores compared to the other two combinations. However, the difference in scores is minimal, indicating similar performance levels.

Table 1. Tacotorn2 + Vocoder MOS Evaluation

	Mean Opinion Score
Tacotron2 + WaveGlow	3.77 ±1.02
Tacotron2 + MelGAN	3.77 ±1.14
Tacotron2 + Multi-band MelGAN	3.87 ±0.96

3.2 Results with FastSpeech2 + Vocoder Combination

The combination with MelGAN consistently maintained good synthesis quality. However, unlike Tacotron2, it lacked intonation, making it dissimilar to human voices. With MelGAN, the synthesis quality was unstable each time. Below, Table 2 shows the evaluation of synthesized sound when combining FastSpeech2 and Vocoder. According to Table 2, the FastSpeech2 + Multi-band MelGAN combination received a high score of 3.83. Meanwhile, the FastSpeech2 + MelGAN combination received a respectable score of 3.29. Consequently, it was found that FastSpeech2 performs well when combined with Multi-band MelGAN.

Table 2. FastSpeech2 + Vocoder MOS Evaluation

	Mean Opinion Score
FastSpeech2 + MelGAN	3.29 ±1.08
FastSpeech2 + Multi-band MelGAN	3.83 ±0.91

Fundamentally, a MOS score above 3 is considered satisfactory for listening. As seen in Tables 1 and 2, all combinations exceeded a MOS score of 3, indicating respectable performance. For example, Tacotron2 showed good performance in combination with Multi-band MelGAN, and FastSpeech2 also demonstrated good performance with Multi-band MelGAN. On the other hand, Tacotron2 received a similar score with MelGAN as with Multi-band MelGAN, indicating it also performs well. However, in FastSpeech2, the combination with MelGAN did not score as well as with Multi-band MelGAN, suggesting it has poorer performance compared to Multi-band MelGAN. This indicates that each Text-to-Mel model has a suitable Vocoder..

4. Conclusion

This paper conducted research to develop methods for generating good sound quality by training and combining various Text-to-Mel and Vocoder models using the LJ Speech Dataset. The results showed that Tacotron2, when combined with Multi-band MelGAN, and FastSpeech2, when combined with Multi-band MelGAN, both received favorable outcomes. This research suggests that specific combinations of Text-to-Mel and Vocoder can create a TTS system that produces high-quality synthesized speech. Furthermore, such combinations could provide auditory pleasure by offering human-like voices to visually imapired users of TTS systems, rather than synthetic sounds.

Acknowlegement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1F1A1062953).

Refrences

[1] Dosik Moon, “Development and Evaluation of a English Speaking Task Using Smartphone and Text-to-Speech”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) Vol. 16, No. 5, pp.13-20, Oct, 31, 2016.

[2] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, Rif A. Saurous, “Tacotron: Towards End-to-End Speech Synthesis”DOI: https://arxiv.org/abs/1703.10135

[3] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, Yonghui Wu, “Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions”DOI: https://arxiv.org/abs/1712.05884

[4] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, Tie-Yan Liu, “FastSpeech 2: Fast and High-Quality End-to-End Text to Speech”DOI: https://arxiv.org/abs/2006.04558
[5] Ryan Prenger, Rafael Valle, Bryan Catanzaro, “WaveGlow: A Flow-based Generative Network for Speech Synthesis”DOI: https://arxiv.org/abs/1811.00002
[6] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio, Aaron Courville, “MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis”DOI: https://arxiv.org/abs/1910.06711
[7] Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, Lei Xie, “Multi-band MelGAN: Faster Waveform Generation for High-Quality Text-to-Speech”DOI: https://arxiv.org/abs/2005.05106