Text-to-speech Synthesis System based on Wavenet

Yuan Li
yuanli92 †

Xiaoshi Wang
xiaoshiw †

Shutong Zhang
zhangst †

Abstract

In this project, we focus on building a novel parametric TTS system. Our model is based on WaveNet(Oord et al, 2016), a deep neural network introduced by DeepMind in late 2016 for generating raw audio waveforms. It is fully probabilistic, with the predictive distribution for each audio sample conditioned on all previous samples. The model introduces the idea of convolutional layer into TTS task to better extract valuable information from the input data. Because the results of our system are not satisfactory, the defects and problems in the system are also discussed in this paper.

1 Introduction

Humans' most natural form of communication, speech, is one of the most difficult approaches to be understood by machines. Text-to-speech(TTS) is a type of Speech synthesis that converts language text into speech, which is mostly driven by engineering efforts to improve above research. TTS has lots of benefits such as speeding up human-computer interaction process and helping hearing impaired people. It is being actively researched nowadays building high quality synthetic voices based on the inputting speech data. In our project we focus on building a novel parametric TTS system.

TTS conversion involves converting a stream of text into a speech waveform. This conversion process largely includes the conversion of a phonetic representation of the text into a number of speech parameters. The speech parameters are then converted into a speech waveform by a speech synthesizer(Dutoit, 1997). The problem with synthesis-by-rule(Mattingly, 1981) systems is that the transitions between phonetic representations are not natural due to transition rules tend to produce only a few styles of transition. In addition, a large set of rules must be stored. Another major problem with the traditional synthesis techniques is the robotic speech quality.

Hidden Markov Model (HMMs) and Deep Neural Network (DNN) are two main approaches for acoustic modeling. WaveNet(Oord et al, 2016), a DNN for generating raw audio waveforms, which yields the cutting edge performance. When applied to TTS, its highly rated by human listeners, which generates sounds significantly more natural than the best parametric for both English and Mandarin. In this project, we aim to build up the model as waveNet. We are able to generate murmurs from our trained models. After parameter tuning and training, our generated speeches are still not very clear and can not achieve the quality of opensource samples from Google. The defects and problems in the system are discussed in this paper.

2 Background

While significant research efforts, from engineering, to linguistic, and to cognitive sciences, have been spent on improving machines' ability to understand speech. Gnerating speech with computer, is still largely based on concatenative TTS(Schwarz, 2005), a technique for synthesising sounds by concatenating short samples of recorded sound and then reorganized to form complete utterance.

This makes it difficult to modify the voice without recording a whole new database, which

†@stanford.edu

has led to a strong demand for parametric TTS(King, 2010). All the information required to generate the data is stored in the parameters of the model, and the contents and characteristics of the speech can be controlled via the inputs to the model.

In order to better understand and compare research techniques in building corpus-based speech synthesizers on the same data, the Blizzard Challenge has been devised. For the 2016 Blizzard Challenge, (Juvela et al, 2016) it aims to build a voice based on the audiobook data read by a British English female speaker.Based on NII parametric speech synthesis framework, it uses Long short term Memory(LSTM), and Recurrent Neural Network(RNN) for acoustic modeling. The results show that the proposed system proposed system outperforms the HTS benchmark and ranks similarly with the DNN benchmark. As metioned in the paper, the audiobook data set was challenging for parametric synthesis, partially due to the expressiveness inherent to audiobooks, but also because of the signal level non-idealities affecting vocoding. For data preprocessing, experimenting more with state-of-the-art de-reverberation and noise reduction methods, and applying a more strict speech/non-speech classification, as the audiobook data also contained non-speech signals such as ambient effects.

However, parametric TTS has tended to sound less natural than concatenative. Existing parametric models typically generate audio signals by passing their outputs through signal processing algorithms known as vocoders.The novelty of WaveNet is changing this paradigm by directly modelling the raw waveform of the audio signal, one sample at a time.

3 Approach

3.1 Dataset

CSTR VCTK Corpus is used for our dataset. This corpus includes speech data uttered by 109 native speakers of English with various accents. Each speaker reads out about 400 sentences, most of which were selected from a newspaper plus the Rainbow Passage and an elicitation paragraph intended to identify the speakers accent. The newspaper texts were taken from The Herald (Glasgow). Each speaker reads a different set of the newspaper sentences, where each set was selected using a greedy algorithm designed to maximize the contextual and phonetic coverage.
Following1 is a sample audio file corresponding to the text input *"The rainbow is a division of white light into many beautiful colors."* is shown below:

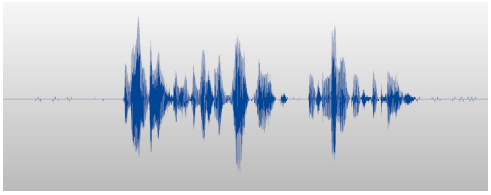

Figure 1: Visualization of sample wav file

We also include the visualization in figure 2 to show all the input data points using PCA to convert them to 3-dimensional space.

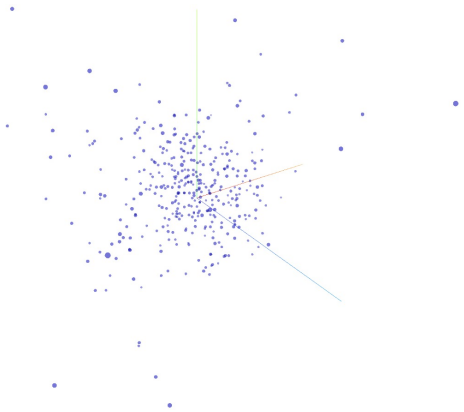

Figure 2: Input feature distribution in 3-dimensional space

3.2 Speech Synthesis System, WaveNet

Introduced by DeepMind in Oct. 2016, WaveNet is a deep learning speech synthesis system using CNN instead of RNN. According to DeepMind, the WaveNet is able to outperform state-of-art HMM-driven unit selection concatenative model and LSTM-RNN based statistical parametric model in subjective paired comparison tests and mean opinion score (MOS) tests.

In general, the idea of WaveNet is to predict the audio sample x_t based on previous samples x_1, ..., x_{t-1}. For TTS task, the network input is audio waveforms(one audio sample per step) plus

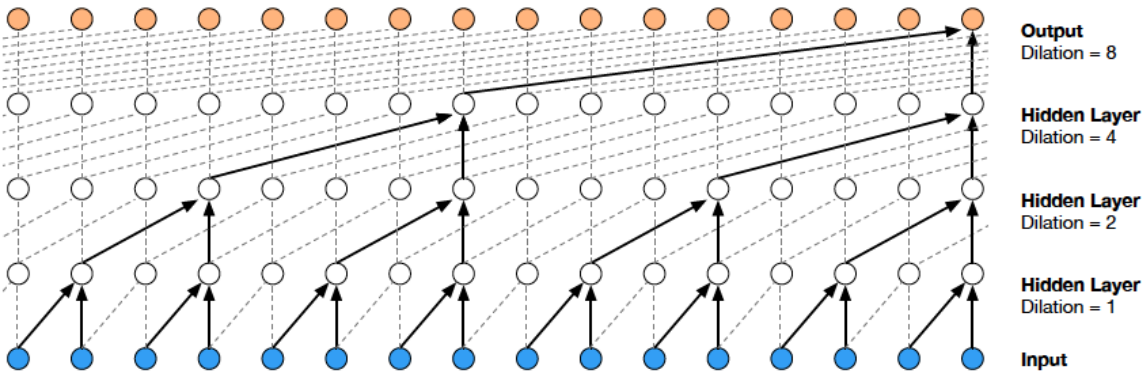

Figure 3: stack of dilated casual convolution layers, figure from (Oord et al, 2016)

linguistic features generated from corresponding texts. The network output at each step $\hat{x_t}$ is the prediction of the audio sample at next time step, given input audio sample from previous steps. The prediction is categorical, namely there is a probability corresponding to each possible next-step audio sample. The loss is the difference between predicted sample and real audio sample in next step. During test, the input is a initial audio sample and linguistic features from test texts. The output for each time step is used as the input for next step. Finally the sequence of output samples is the generated speech for test text.

3.2.1 Dilated Convolution

The structure of WaveNet is based on Dilated Convolution. We can think of the dilated convolution as convolution with filters having holes. The intuition is with dilated convolution, the output of one time step is able to depend on a long sequence of inputs from previous time steps. This is how the network achieve $P(x_t|x_1, ..., x_{t-1})$ in practice. Figure3 provides a visualization of dilated convolution with filter size $1 * 2$. In the figure, we can see that 3 hidden layers are needed to gather information from 16 inputs. Without dilation, we need 15 hidden layers.

3.2.2 Gated Activation and Residual Units

In the nonlinearity part of network structure, Oord et al apply a gated activation unit similar to the activation in LSTM. In detail, the activation formula is the following:

$$z = \tanh(W_{f,k} * x) \times \sigma(W_{g,k} * x)$$

where σ represents sigmoid function, Ws are weights and $\times$ represents element wise multiplication.

In the network structure Oord et al also apply residual net structure(He et al, 2016), which is shown to be useful for making deep network converge.

3.2.3 Local Conditioning and Global Conditioning

Oord et al also introduced the idea of conditioning to include more information in the model in order to achieve more meaningful tasks. Conditioning is implemented through adding more learn-able parameters. Currently there are two kinds of conditioning, global and local. Global conditioning record the information of different speakers such that during generation people can choose to generate voice from a specific speaker. Local conditioning record the text information of training data, so that people can choose to generate audio from a specific text file, which is exactly the goal of TTS. Following is the local conditioning version of activation function:

$$z = \tanh(W_{f,k}*x+V_{f,k}*y)\times\sigma(W_{g,k}*x+V_{g,k}*y)$$

where Vs are newly introduced learnable variables and y is a function character from input text file.

4 Experiments and Analysis

We apply a tensorflow implementation of WaveNet based on ibab's github code for training and testing. We also tested a smaller version of WaveNet with fewer parameters (96160 parameters) using another open source directory from basveeling's github repository. But the results are even worse since the model cannot adopt to

VCTK dataset very well. Currently we have not found any complete implementation of local conditioning on open source community. There is a partially finished(training) version from alexbeloi on github, we changed their software to make the code work without local conditioning and than tried to finish the testing part. In our project, we have not completed local conditioning.

4.1 Experiments with Whole VCTK datasets

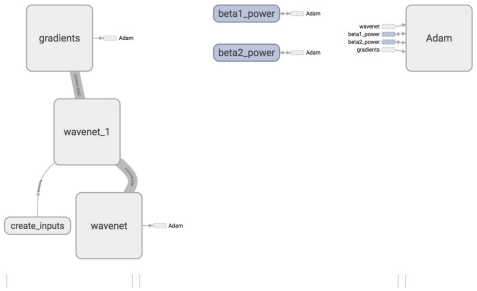

Figure 4: Visualization of the tensorflow version of the implementation for WaveNet

A simple visualization of the model we trained is shown in figure 4.

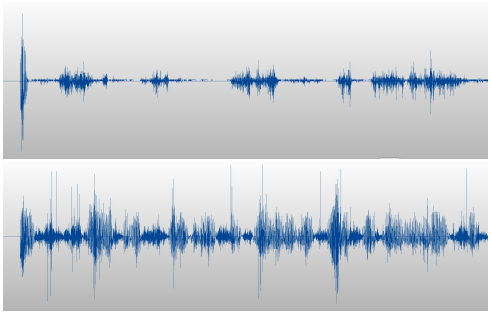

Figure 5: Sample outputs generated from trained model using the whole VCTK dataset

Figure 5 illustrates the output wav files we generated after training WaveNet for 16350 steps. These two audios are typical outputs we got from the trained models using different parameter settings. Some of the trained models also generate pure noise which we think the reason behind may be that these models converge to local optimum. Compared to what the audio input file looks like, the curves for these murmurings show that there are many differences between these with real audio files. The upper audio wav curve shows that we only generate very few pulses with much smaller amplitude. While in the second wav file, the model did not learn 'silence' very well since some syllables have started to shown, but the generated file failed to connect these syllables to words and learn the silences between syllables. We still see the improvements between the white noise generated at the beginning when the loss is around 5. Some basic patterns have already started to shown when we use global conditioning in WaveNet.

4.2 Experiments with data from a single speaker from VCTK datasets

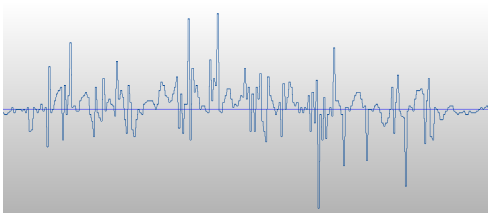

Figure 6: Visualization of wav file generated from tiny dataset model

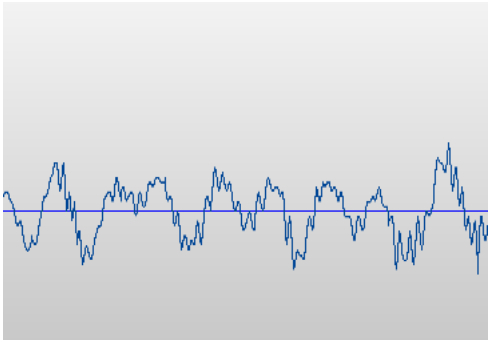

Figure 7: Visualization of wav file generated from whole dataset model

The whole VCTK dataset has around 45000 samples and the output voice samples from Google are generated from a model trained 88000 iterations. For this project we are not able to find enough GPUs to do experiments with a such a large number of iterations. In order to mitigate the limitation of our computational capacity and try to make the deep network become easier to train, we create a tiny dataset based on samples from speaker 339, a female speaker from Philadelphia, America. The tiny dataset contains 423 samples, about 1% of the whole dataset. We trained the network with batch size of 4 and we tried with/without regularization, 3000 iterations(around 30 rounds per sample) each. The table 1 shows the average loss of last 10 iterations.

As we can see, the loss of tiny dataset is less

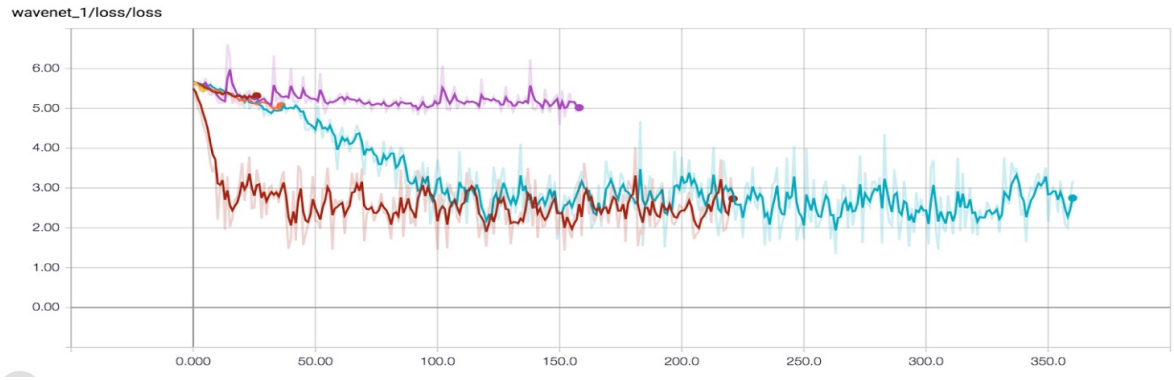

Figure 8: Loss curve during the training process with batch size 10

	No Reg	L2 Reg, weight = 0.01
Training loss	1.663	2.258

Table 1: Comparison of training average loss with or without regularization

than that of large dataset. However, the result is not positive after we generated audio samples using the newly trained model. Figure 6 and 7 show a comparison of clips of generated wav files from model based on tiny dataset and model based on whole dataset. As we can see, model with tiny dataset does show repeated patterns(repeated 3 times in the graph) to some extent, but the patterns are not as clear as the patterns from whole dataset model, which are very close to non-noise voice.

4.3 Analysis

We are able to generate murmurs from our trained models using the whole dataset. For tiny dataset, the results are not as good as whole dataset. After parameter tuning and training, our generated speeches are still not very clear and can not achieve the quality of open source samples from Google.

The relatively successful generated speeches follows the pattern that the amplitude is low, which is different from our unsuccessful(noisy) results that the amplitude is high and frequency changes very quickly. As we can see from figure 8, training loss hits plateau (2.0-3.0) fairly quickly (around 150 iterations) and cannot further goes down with hyperparamter tuning (for example, applying lower learning rates or using different batch sizes). Moreover, number of iterations matters even if loss hits plateau. We find that with small number of iterations, the voice of generated audios quickly come to a silent stage(with little fluctuation in voice). Longer the iterations, slower this happens.

The possible reasons for the relatively low quality of generated speeches are the following:

1. Number of training iterations are not enough(we used 16350 iterations without global condition and 5100 iterations with global iterations, comparing to around 80000 iterations from Google.)

2. Very careful parameter tuning is needed to achieve a good model with the whole training corpus due to the large number of samples (>45000) and variations in accents and gender. Since DeepMind's paper didn't mention how they set all the parameters, thus we have to randomly guess them. We also suspect maybe other tricks need to be apply to the training process as well.

4.4 Evaluation

Based on our exploration of this domain, it is very difficult to give quantitative analysis on the performance of a speech synthesis system. The general evaluation method proposed by state-of-art research works (Chang et al, 2011) is to use MOS (Mean Opinion Score) during listening tests. Moreover, the resulting outputs would be compared to the results generated by baseline methods and feedbacks will be collected afterwards. Since crowdsourcing is not applicable to our class project, we only test the outputs within the three of us. The output wav files only contain murmuring thus it is very straightforward to tell which one has the better quality.

5 Conclusion and Future Works

In our project, we introduced a newly published method WaveNet into the task of Text-to-speech. After reviewing open source code from the web, we explored some incomplete versions to obtain output audio files by tuning the hyperparameters. The results shown above showed that WaveNet model have already learned some basic patterns without any information of the text. But we are unable to get similar results compared to output samples published by DeepMind. In the experimental result part, we include error analysis about the reasons about why we generate these results and some possible ways to better reduce the total loss.

As for our future work, we can explore from these few aspects,

1. Train models based on global conditioning with reasonable outputs. Based on the analysis we did in section 4, with better computational resources we have a lot trick that we could apply to the training process.

2. Finish local conditioning based on open-source code base.

3. Since DeepMind didn't release its dataset, maybe we could manually tailor VCTK dataset to improve the quality of these audio files to better boost the performance of WaveNet model.

4. One of the ideas we had was to explore filter size in convolutional layer to see its influence on the output. The size was set to 2 in the WaveNet model and we wish to see what the outputs would be if we increase the size.

References

Aaron van den Oord and Sander Dieleman and Heiga Zen and Karen Simonyan and Oriol Vinyals and Alex Graves and Nal Kalchbrenner and Andrew Senior and Koray Kavukcuoglu *WaveNet: A Generative Model for Raw Audio* arXiv:1609.03499

Diemo Schwarz *Current research in concatenative sound synthesis* International Computer Music Conference (ICMC)

Simon King *A beginners guide to statistical parametric speech synthesis* cit

Juvela, Lauri and Wang, Xin and Takaki, Shinji and Kim, SangJin and Airaksinen, Manu and Yamagishi, Junichi *The NII speech synthesis entry for Blizzard Challenge 2016*

Zhao, Yi and You, Xiu and Saito, Daisuke and Minematsu, Nobuaki *The UTokyo System for Blizzard Challenge 2016*

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun *Deep Residual Learning for Image Recognition* arXiv:1512.03385

Chang, Yu-Yun *Evaluation of TTS systems in intelligibility and comprehension tasks*

Thierry Dutoit *An introduction to text-to-speech synthesis* Springer Science & Business Media

Ignatius G Mattingly *Phonetic representation and speech synthesis by rule* The cognitive representation of speech, 415-420 North Holland Amsterdam