The Impact of Deep Learning on Speech Synthesis with Embedded or Mobile Devices

Hannes Bohnengel
Technical University of Munich
hannes.bohnengel@tum.de

ABSTRACT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

KEYWORDS

Deep Learning, Deep Neural Network, Embedded System, Mobile Device, Speech Synthesis, Text-to-Speech

1 INTRODUCTION

Virtual personal assistants (VPA) like Siri, Cortana or Google Now start having a huge impact on the way of interacting with electronic devices like smartphones or notebooks. Up to now the VPAs help only with rather simple tasks like search queries, starting phone calls or setting a clock, but according to a recent survey from the IT research firm Gartner [3], this will change in the near future. With the Facebook Messenger it is already possible to make purchases or to order an Uber car and new use cases are expected soon. The survey also states, that through the vast increase of devices in the scope of the Internet of Things (IoT) the way of interacting with machines will go towards minimal or zero touch. Instead of interacting through common touch-displays or buttons, the user simply speaks to the device, like to another person. To enable this, both Automatic Speech Recognition (ASR) and speech synthesis are essential technologies.

In this paper I will only focus on the speech synthesis part. A widely spread technique to synthesize human speech from a given text or from linguistic descriptions is Statistical Parametric Speech Synthesis (SPSS); also referred to as Statistical Parametric Speech Generation (SPSG) [8]. This technique is based on the usage of Hidden Markov Models (HMMs). Zen *et al.* [13] show that it has several advantages over its predecessor, the concatenative speech synthesis, for example the flexibility in changing voice characteristics and a smaller memory footprint. However the quality of the generated speech still has potential for improvement. Due to over-smoothing the voice sounds muffled in comparison to natural speech.

This is where recent achievements in deep learning come in. Deep learning is usually referred to as a class of machine learning techniques that achieve tasks like feature extraction or pattern analysis by using many connected layers of non-linear information processing [5, 8]. Since 2006 advances in the training algorithms of Deep Neural Networks (DNNs) have enabled the field of deep learning applications to emerge [4]. Most machine learning models until then had used shallow structures, like for example HMMs, Gausian Mixture Models (GMMs), Conditional Random Fields (CRFs) or Support Vector Machines (SVMs). In these structures only one layer is responsible for generating features out of the raw input signals. While achieving quite good results with rather simple problems, they reach their limit when it comes to more complex tasks like processing human language or natural images [5]. In the tutorial survey [5] the author also states four different approaches to improve speech synthesis through deep learning models, whereof three are dealing with SPSS. One of those three approaches is described in [12], where the authors implemented a part of the speech synthesis system by using a DNN and observed an improved performance in predicting output features. In [6] a more general approach is conducted by investigating what effects the deployment of a DNN on different parts of the SPSS system has. An improvement of the naturalness of the generated speech was one of the main results.

For implementing speech synthesis on resource-constrained devices like smarthpones or tablets, SPSS is considered the best solution due to the tradeoff between voice quality and acceptable footprint size [10]. Since the computational costs of SPSS are often high, some optimization steps like applying fewer conditional calls are conducted in [10] to make the HMM-based speech synthesis technique SPSS more suitable for mobile devices. Going one step further, in [4] an approach to adapt SPSS for embedded devices by using a deep learning model, an Auto Encoder (AE), is employed. Four tasks (syllabification, phonetic transcription, part-of-speech tagging and lexical stress prediction) are examined and tested with the use of this deep learning model. As results the authors highligh hugly reduced model sizes, higher training times, very close performance and a similar run time in comparison to the state-of-the-art models. This shows that the usage of deep learning models for speech synthesis on embedded systems is a reasonable step, not only to improve performance and voice quality, but also towards the independability on online databases for speech synthesis.

The remaining paper is structured as follows: Section 2 first states the motivation, why speech synthesis is a useful technology. Then it describes the conventional approach without deep learning models for speech synthesis and gives an overview of advantages and drawbacks of the used models and techniques. This is followed by a brief explanation of the probably most common used technique SPSS, where the paper [13] has been chosen as commonly cited reference. Thereafter two possibilities how SPSS can be improved by deploying deep learning models are characterized, wherefore the papers [6, 12] are reviewed. In Section 4 the motivation, why speech synthesis is important on embedded or mobile devices is given, followed by two examples on how speech synthesis can be implemented on an embedded system, once without [10] and once with deep learning models [4]. Finally Section 5 summarizes the essential points of this paper and gives some future directions.

2 CONVENTIONAL SPEECH SYNTHESIS

2.1 Motivation & Approaches

Speech synthesis has emerged over the last 10 years due to a vast contribution by the global community of researchers and the increasing computational power for data processing. Its quality and naturalness has increased steadily and different approaches have been developed so far [9]. The typical applications like navigation systems in cars or telephone-based dialogue systems are nowadays widely established. But also as reading aid for visually impaired people [1] or as in the case of the famous scientist Stephen Hawking, who has been using a synthesized voice to communicate since 1997 [2], speech synthesis has proven to be very useful. Another very interesting application of speech synthesis is shown in [XX]. The author proposed to introduce synthetic speech as means of communication between pilots, since there have been many accidents due to missunderstandings at radio-based communication.

According to [7] speech synthesis can be divided into three types: Canned speech, Context-to-Speech (CTS) and Text-to-Speech (TTS). Canned speech more or less is the replay of prerecorded spoken sentences or words with none or very little adjustments. A typical example are the announcements on train stations. Because of the high effort of recording everything (almost) exactly as it is replayed, this approach is limited to only a few simple applications. With CTS the waveform is generated out of a linguistic description without any information of the respective text. In this way no natural language processing is required, but nevertheless CTS nowadays has not made any important impact. The last and most promising type is TTS.

A TTS system consists of a Natural Language Processing (NLP) part, where the text is analysed and the word and sentence structure and accents are extracted. In the next step these accents are used to generate the prosody of the given text like duration, intensity and pitch. Then the created phonetic represantations with prosody information are stringed together to a continuous stream of signal parameters. The last task, the speech generation, uses this stream to generate the respective waveform. This function block can be implemented in different ways. In [7] three general approaches are named as follows: Parametric Speech Synthesis (formant-based synthesis), Concatenative Speech Synthesis (unit-selection synthesis) and Statistical Parametric Speech Synthesis (HMM-based synthesis). The methods in brackets are the respective implementations, which are most commonly used.

The formant-based synthesis is the oldest approach. To generate a voice waveform, an excitation signal is fed into multiple formant filters which describe the characteristics of the human vocal tract. The output of the filters then forms the voice waveform. This technique is the only one which does not need any recorded speech, but instead generates the synthesized voice only by modeling the human vocal tract. The quality of the generated voice is the lowest in comparison to the other techniques, but therefore formant-based synthesizer have the smallest footprint and the voice characteristics can easilily been modified by just changing the filter parameters [7].

With the development of concatenative speech synthesizer the quality of the generated speech improved tremendously. Very similar to CTS prerecorded speech is used as reference. Very basically said, the recorded speech is devided into units and these units are then stringed together to form the new speech signal according to a given text. Hereby the chosen size of the units determines both the footprint size and the voice quality. With larger units a higher voice quality can be achieved, but this also results in a much larger database. The challenge in unit-selection is to ensure, that the transissions between the units are as natural as possible [7]. In Section 2.2 some concepts on how to achieve this as well as the third implementation, the HMM-based synthesis, a specific instance of SPSS are described in detail.

2.2 HMM-based Speech Synthesis

In this section the HMM-based, the most recent approach for speech synthesis will be described further and both advantages and drawbacks compared to unit-selection synthesis will be highlighted. Therefore the work of Zen et al. [13] will be taken as reference.

The quality of unit-selection synthesis directly depends from the quality of the prerecorded speech. But even with a database of excellent quality sporadic errors can still not be avoided totally. If a specific phonetic or prosodic part of a generated sentence is not well represented in the database the output quality of this sentence suffers immensely. To try to avoid this a huge effort in specifically designing the database for the required application can be performed, but still there is no guarantee that such bad joins happen. In addition the fact, that in unit-selection no or only very little adaptions of the voice characteristics are possible without an enormous increase of the database size, the ambition towards seamless speech synthesis leads to the HMM-based approach. There a statistic representation of some sets of speech segments is used to generate arbitrary synthetic speech.

Details about unit-selection synthesis ???

In Figure 1 the structure of a typical HMM-based synthesizer is shown. The whole system can be devided into two parts, the training and the synthesis part. The connection of these two parts are a number of context-dependent (what is that?) HMMs. In the training part spectrum and excitation parameters of the recorded speech are used to generate acoustic models represented by the HMMs. Thereby phonetic, linguistic and prosodic parameters are considered. Details about spectrum and excitation parameters? In comparison to a unit-selection system the database only is needed in the training part.

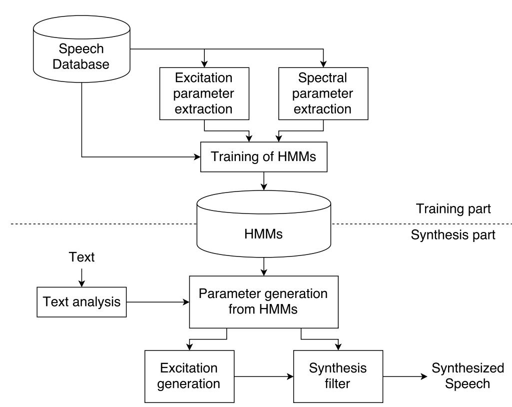

Figure 1: Function blocks of HMM-based synthesis [13]

In the synthesis part first the text which is to be synthesized is

The main disadvantage of this approach compared to unit-selection synthesis is the quality of the synthesized speech. Three factors are accountable for the lack of quality: vocoder, modeling accuracy, and over-smoothing.

In Table 1 these techniques are compared regarding the most prominent advantages and drawbacks.

Table 1: Comparison of speech generation methods [7, 13]

Technique	Advantages	Drawbacks
Formant-based	No prerecorded speech required	Very artificial and metallic voice
Unit-selection	Very high voice quality possible	Large database required
HMM-based	Adjustable voice and small footprint	Voice sounds muffled

Why there is need to further improve this technology?
What is a HMM?

3 SPSS WITH DEEP LEARNING MODELS

3.1 General ways for improvement

In [6] the effects of deep learning methods on SPSS are investigated. Therefore the different parts of a HMM-based speech synthesis system are modeled with a DNN and then the output is compared to the conventional approach.

3.2 One specific approach for improvement

Statistical parametric speech synthesis using deep neural networks [12]

4 SPEECH SYNTHESIS ON EMBEDDED DEVICES

4.1 Motivation

Why is it important to implement speech synthesis on embedded platform?
What needs to be thought about when dealing with embedded or mobile devices?

4.2 HMM-based Approach

An example of how speech synthesis can be implemented on embedded platform without deep learning (core paper 3).

4.3 Deep Learning-based Approach

An example of how speech synthesis can be implemented on embedded platform WITH deep learning (core paper 4).

5 CONCLUSIONS

Here the core points will be repeated and concluded.
Some future aspects will be highlighted.
What should be done in the future?

See [11]

- Voice cloning
- Voice reconstruction
- Personalised speech-to-speech translation
- Articulatory-controllable speech synthesis

REFERENCES

[1] 2017. The Benefits of Text to Speech. (02 June 2017). http://www.readspeaker.com/benefits-of-text-to-speech/

[2] 2017. Stephen Hawking - The Computer. (02 June 2017). http://www.hawking.org.uk/the-computer.html

[3] 2017. Survey on usage of virtual personal assistants by Gartner. (22 May 2017). http://www.gartner.com/newsroom/id/3551217

[4] Tiberiu Boroş and Stefan Daniel Dumitrescu. 2015. Robust Deep-learning Models for Text-to-speech Synthesis Support on Embedded Devices. In *Proceedings of the 7th International Conference on Management of Computational and Collective intElligence in Digital EcoSystems (MEDES '15)*. ACM, New York, NY, USA, 98–102. https://doi.org/10.1145/2857218.2857234

[5] Li Deng. 2014. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing* 3 (January 2014). https://doi.org/10.1017/atsip.2013.9

[6] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda. 2015. The effect of neural networks in statistical parametric speech synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 4455–4459. https://doi.org/10.1109/ICASSP.2015.7178813

[7] Florian Hinterleitner. 2017. *Quality of Synthetic Speech: Perceptual Dimensions, Influencing Factors, and Instrumental Assessment (T-Labs Series in Telecommunication Services)* (1st ed. 2017 ed.). Springer, Singapore. https://doi.org/10.1007/978-981-10-3734-4

[8] Z. H. Ling, S. Y. Kang, H. Zen, A. Senior, M. Schuster, X. J. Qian, H. M. Meng, and L. Deng. 2015. Deep Learning for Acoustic Modeling in Parametric Speech Generation: A systematic review of existing techniques and future trends. *IEEE Signal Processing Magazine* 32, 3 (May 2015), 35–52. https://doi.org/10.1109/MSP.2014.2359987

[9] David Suendermann, Harald Höge, and Alan Black. 2010. *Challenges in Speech Synthesis*. Springer US, Boston, MA, 19–32. https://doi.org/10.1007/978-0-387-73819-2_2

[10] Bálint Tóth and Géza Németh. 2012. Optimizing HMM Speech Synthesis for Low-Resource Devices. In *Journal of Advanced Computational Intelligence and Intelligent Informatics*. 327–334.

[11] Junichi Yamagishi. 2011. New and emerging applications of speech synthesis. (February 2011). www.cstr.ed.ac.uk

[12] Heiga Zen, Andrew Senior, and Mike Schuster. 2013. Statistical parametric speech synthesis using deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 7962–7966. https://doi.org/10.1109/ICASSP.2013.6639215

[13] Heiga Zen, Keiichi Tokuda, and Alan W. Black. 2009. Statistical parametric speech synthesis. (April 2009), 1039 - 1064 pages. https://doi.org/10.1016/j.specom.2009.04.004

Remarks starting here:

Page count estimation:

- Section 1 — 1 (incl. Abstract)
- Section 2.1 — 0.75
- Section 2.2 — 0.5
- Section 3.1 — 0.5
- Section 3.2 — 0.5
- Section 4.1 — 0.25
- Section 4.2 — 0.75
- Section 4.3 — 0.75
- Section 5 — 0.25
- References — 0.5

Total: **6 pages ???**

To do:

- cite both long and short version of [13]
- check title
- check references (bibtex warnings ???)
- check correct citing
- check for typos
- check page count (6 pages)
- disable sreen mode on last draft
- explain acronyms first time they appear in abstract AND first time they appear in body

Questions:

(1) Why speech synthesis is important? What are its applications?
(2) What are the conventional techniques of speech synthesis? What are the drawbacks of such techniques?
(3) What is deep learning? What improvements do deep learning algorithms bring?
(4) How some algorithms are modified to suit speech synthesis?
(5) Why is it important to implement speech synthesis on embedded platform?
(6) An example of how speech synthesis can be implemented on embedded platform without deep learning.
(7) How the 3 can be combined?
(8) Future works.