# Deep Recurrent Neural Networks in Speech Synthesis Using a Continuous Vocoder

Mohammed Salah Al-Radhi[1](✉), Tamás Gábor Csapó[1,2],
and Géza Németh[1]

[1] Department of Telecommunication and Media Informatics,
Budapest University of Technology and Economics, Budapest, Hungary
{malradhi, csapot, nemeth}@tmit.bme.hu
[2] MTA-ELTE Lendület Lingual Articulation Research Group,
Budapest, Hungary

**Abstract.** In our earlier work in statistical parametric speech synthesis, we proposed a vocoder using continuous F0 in combination with Maximum Voiced Frequency (MVF), which was successfully used with a feed-forward deep neural network (DNN). The advantage of a continuous vocoder in this scenario is that vocoder parameters are simpler to model than traditional vocoders with discontinuous F0. However, DNNs have a lack of sequence modeling which might degrade the quality of synthesized speech. In order to avoid this problem, we propose the use of sequence-to-sequence modeling with recurrent neural networks (RNNs). In this paper, four neural network architectures (long short-term memory (LSTM), bidirectional LSTM (BLSTM), gated recurrent network (GRU), and standard RNN) are investigated and applied using this continuous vocoder to model F0, MVF, and Mel-Generalized Cepstrum (MGC) for more natural sounding speech synthesis. Experimental results from objective and subjective evaluations have shown that the proposed framework converges faster and gives state-of-the-art speech synthesis performance while outperforming the conventional feed-forward DNN.

**Keywords:** Deep learning · LSTM · BLSTM · GRU · RNN

## 1 Introduction

Statistical parametric speech synthesis (SPSS) based text-to-speech (TTS) systems have steadily advanced in terms of naturalness during the last two decades. Even though the quality of synthetic speech is still unsatisfying, the benefits of flexibility, robustness, and control denote that SPSS stays as an attractive proposition. One of the most important factors that degrade the naturalness of the synthesized speech is known as the limited capabilities of the acoustic model which captures the complex and nonlinear relationship between linguistic and acoustic features [1]. Although there have been many attempts to create a more accurate acoustic model for SPSS (such as [2]), the hidden Markov model (HMM) has been the most popular attempt for a long time [3]. Even though this model can enhance accuracy and synthesis performance, it usually increases the amount of computational complexity with higher number of model parameters [4].

© Springer International Publishing AG 2017
A. Karpov et al. (Eds.): SPECOM 2017, LNAI 10458, pp. 282–291, 2017.
DOI: 10.1007/978-3-319-66429-3_27

Recently, deep learning algorithms have shown their ability to extract high-level, complex abstractions and data representations from large volumes of supervised and unsupervised data [5], and achieve significant improvements in various machine learning areas. Neural approaches have been also used in SPSS as deep feed-forward neural networks (DNNs) with more than one layer of hidden units between its input and output layers. By mapping directly linguistic features to vocoder parameters, DNNs can be viewed as a replacement for the decision tree used in HMM-TTS systems [6]. DNNs have also other advantages, including the ability to model high-dimensional acoustic parameters [7], and the availability of multi-task learning [8]. However, Zen, et al. [9] comprehensively listed the limitations of the conventional DNN-based acoustic modeling for speech synthesis, e.g. its lack of ability to predict variances, unimodal nature of its objective function, and the sequential nature of speech is ignored because it assumes that each frame is sampled independently. In other words, the mapping is performed frame by frame without considering contextual constraints between statics and deltas during training.

To overcome these problems, recurrent neural networks (inserting cyclical connections in feed-forward DNNs) have proven to have an advantage in modeling sequences whose activation at each time is dependent on that of the previous time to shape prediction output. Although it is difficult to train RNNs to capture long term dependencies [10], successful approaches were used to reduce the negative impacts of this limitation. It was suggested in [11] to store information over long or short time intervals to include contextual constraints called as a long short-term memory (LSTM). In [12] a bidirectional LSTM based RNN was employed in which there is a feedback to retain previous states. In [13], gated recurrent unit (GRU) based RNN was proposed to adaptively capture dependencies of different time scales. In this paper, four variants of neural networks are investigated in the speech synthesis scenario and implemented using a continuous vocoder.

In our earlier work, we proposed a computationally feasible residual-based vocoder [14], using a continuous F0 model [15], and Maximum Voiced Frequency (MVF) [16]. In this method, the voiced excitation consisting of pitch synchronous PCA residual frames is low-pass filtered and the unvoiced part is high-pass filtered according to the MVF contour as a cutoff frequency. The approach was especially successful for modelling speech sounds with mixed excitation. However, we noted that the unvoiced sounds are sometimes poor due to the combination of continuous F0 and MVF. In [17], we removed the post-processing step in the estimation of the MVF parameter and thus successfully modelled the unvoiced sounds with our continuous vocoder, which was integrated into a HMM-TTS system. In [18], we successfully modelled all vocoder parameters (continuous F0, MVF, and MGC) with feed-forward DNNs. The goal of this paper is to extend modeling of our continuous vocoder parameters using RNN, LSTM, BLSTM, and GRU models. Besides, noise components in voiced sounds are parameterized and modeled to meet the requirements of high sound quality.

The rest of this paper is structured as follows: Sect. 2 describes the novel methods we used for speech synthesis. Then, experimental conditions are showed in Sect. 3. Evaluation and discussion are presented in Sect. 4. Finally, Sect. 5 concludes the contributions of this paper.

# 2 Methodology

## 2.1 Continuous Vocoder

For the current RNN-TTS experiments, the improved version of our continuous vocoder was used [17]. During the analysis phase, F0 is calculated on the input waveforms by the open-source implementation[1] of a simple continuous pitch tracker [15]. In regions of creaky voice and in case of unvoiced sounds or silences, this pitch tracker interpolates F0 based on a linear dynamic system and Kalman smoothing. Next, MVF is calculated from the speech signal using the MVF_Toolkit[2], resulting in the MVF parameter [16]. In the next step 60-order Mel-Generalized Cepstral analysis (MGC) [19] is performed on the speech signal with alpha = 0.58 and gamma = 0. In all steps, 5 ms frame shift is used. The results are the F0cont, MVF and the MGC parameter streams. Finally, we perform Principal Component Analysis (PCA) on the pitch synchronous residuals [14].

During the synthesis phase, voiced excitation is composed of PCA residuals overlap-added pitch synchronously, depending on the continuous F0. After that, this voiced excitation is lowpass filtered frame by frame at the frequency given by the MVF parameter. In the frequencies higher than the actual value of MVF, white noise is used. Voiced and unvoiced excitation is added together. Finally, an MGLSA filter is used to synthesize speech [20].

### 2.1.1 Improved Version of Continuous Vocoder

In the standard continuous vocoder, there is a lack of voiced components in higher frequencies. However, it was shown that in natural speech, the high-frequency noise component is time-aligned with the pitch periods. For this reason, in a recent study, we applied several time envelopes to shape the high-frequency noise excitation component [21]. From the several envelopes investigated, the True envelope was found to be the best. Therefore, this will be used in the current study.

The True Envelope (TE) algorithm starts with estimating the cepstrum and updating it in such a way that the original spectrum signal and the current cepstral representation is maximized [22]. To have an efficient real time implementation, [23] proposed a concept of a discrete cepstrum which consists of a least mean square approximation, and [24] added a regularization technique that aims to improve the smoothness of the envelope. Here, the procedure for estimating the TE is shown in Fig. 1 in which the cepstrum can be calculated as the inverse Fourier transform of the log magnitude spectrum of the voiced frame. Moreover, TE with weighting factor will bring us a unique time envelope which makes the convergence closer to natural speech. In practice, the weight factor which was found to be the most successful is 10.

[1] https://github.com/idiap/ssp.

[2] http://tcts.fpms.ac.be/ ~ drugman/files/MVF.zip.

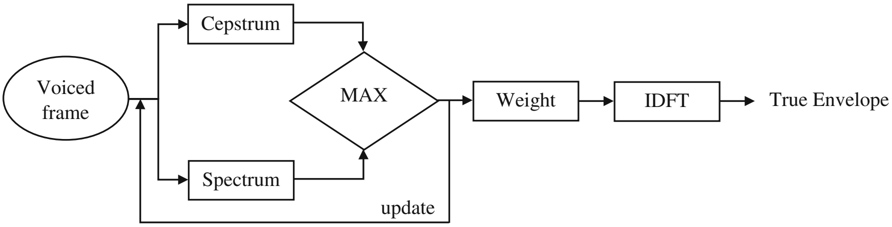

**Fig. 1.** Procedures for estimating the true envelope

## 2.2 Acoustic Modeling

Figure 2 conceptually illustrates the main components of the continuous vocoder when applied in RNN-based training. Textual and phonetic parameters are first converted to a sequence of linguistic features as input, and neural networks are employed to predict acoustic features as output for synthesizing speech. Because standard RNNs with sigmoid activation function suffer from both vanishing gradients and exploding [10], our goal is to present and evaluate the performance of recently proposed recurrent units on sequence modeling for improved training of the continuous vocoder parameters.

### 2.2.1 Feedforward DNN (Baseline)

DNNs have become increasingly a common method for deep learning to achieve state-of-the-art performance in real-world tasks [6, 8]. Simply, the input is used to predict the output with multiple layers of hidden units, each of which performs a non-linear function of the previous layer's representation, and a linear activation function is used at the output layer. In this paper, we use our baseline model [18] as a DNN with feed-forward multilayer perceptron architecture. We applied a hyperbolic tangent activation function whose outputs lie in the range (−1 to 1) which can yield lower error rates and faster convergence than a logistic sigmoid function (0 to 1).

### 2.2.2 Recurrent NN

A more popular and effective acoustic model architecture is a version of the recurrent neural networks (RNNs) which can process sequences of inputs and produces sequences of outputs [13]. In particular, the RNN model is different from the DNN the following way: the RNN operates not only on inputs (like the DNN) but also on network internal states that are updated as a function of the entire input history. In this case, the recurrent connections are able to map and remember information in the acoustic sequence, which is important for speech signal processing to enhance prediction outputs.

### 2.2.3 Long Short-Term Memory

As originally proposed in and recently used for speech synthesis [25], long short-term memory networks (LSTM) are a class of recurrent networks composed of units with a particular structure to cope better with the vanishing gradient problems during training and maintain potential long-distance dependencies [11]. This makes LSTM applicable to learn from history in order to classify, process and predict time series. Unlike the conventional recurrent unit which overwrites its content at each time step, LSTM have

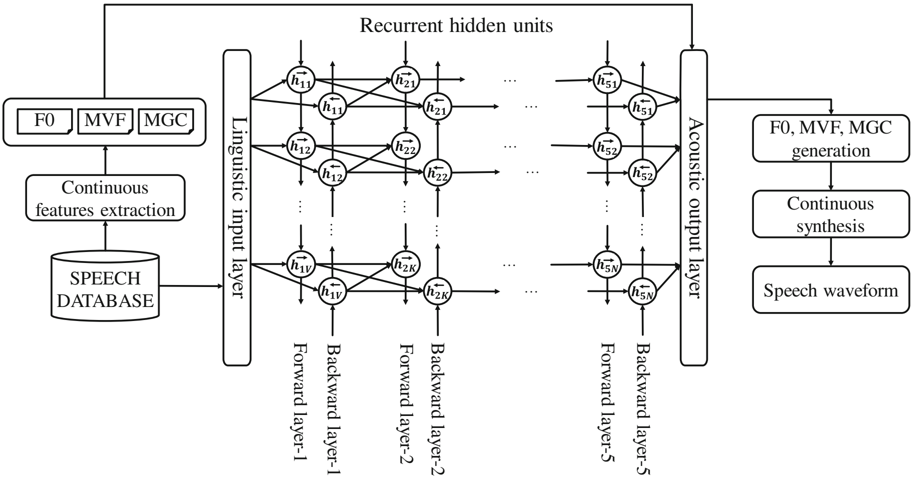

**Fig. 2.** A general schematic diagram of the proposed method based on recurrent networks

a special memory cell with self-connections in the recurrent hidden layer to maintain its states over time, and three gating units (input, forget, and output gates) which are used to control the information flows in and out of the layer as well as when to forget and recollect previous states.

### 2.2.4 Bidirectional LSTM

In a unidirectional RNN (URNN) only contextual information from past time instances are taken into account, whereas a bidirectional RNN (BRNN) can access past and future contexts by processing data in both directions [26]. BRNN can do this by separating hidden layers into forward state sequence and backward state sequence. Combining BRNN with LSTM gives a bidirectional-LSTM (BLSTM) which can access long range context in both input directions, and can be defined generally as in [12].

### 2.2.5 Gated Recurrent Unit

A slightly more simplified variation of the LSTM, the gated recurrent unit (GRU) architecture was recently defined and found to achieve a better performance than LSTM in some cases [13]. GRU has two gating units (update and reset gates) to modulate the flow of data inside the unit but without having separate memory cells. The update gate supports the GRU to capture long term dependencies like that of the forget gate in LSTM. Moreover, because an output gate is not used in GRU, the total size of GRU parameters is less than that of LSTM, which allow that GRU networks converge faster and avoid overfitting.

## 3 Experimental Conditions

### 3.1 Data

To measure the performance of the obtained model, the US English female (SLT) speaker was chosen for the experiment from the CMU-ARCTIC database [27], which consists of 1132 sentences. 90% of the sentences were used for training and the rest was used for testing.

### 3.2 Network Topology and Training Settings

Neural network models used in this research were implemented in the Merlin open source speech toolkit[3] [25]. For simplicity, the same architecture is used in both duration and acoustic models. Weights and biases were prepared with small nonzero values, and optimized with stochastic gradient descent to minimize the mean squared error between its predictions and acoustic features of the training set. The Speech Signal Processing Toolkit [28] was used to apply the spectral enhancement. Delta and delta-delta features were calculated for all the features. The input linguistic features have min-max normalization, while output acoustic features have mean-variance normalization. In general, the design configuration of current neural network model is similar to those we have given in [18]. The training procedures were conducted on a high performance NVidia Titan X GPU.

We trained a baseline DNN and four different recurrent neural network architectures, each having either LSTM, BLSTM, GRU, or RNN. Each model has fairly the same number of parameters, because the objective of these experiments is to compare all four units equally in order to find out the best unit to model our continuous vocoder. The systems we implemented are as follows:

- **DNN:** This system is our baseline approach [18] which uses 6 feed-forward hidden layers; each one has 1024 hyperbolic tangent units.
- **LSTM:** 4 feed-forward hidden lower layers of 1024 hyperbolic tangent units each, followed by a single LSTM hidden top layer with 512 units. This recurrent output layer makes smooth transitions between sequential frames while the 4 bottom feed-forward layers intended to act as feature extraction layers.
- **BLSTM:** Similar to the LSTM, but replacing the LSTM top layer with a BLSTM layer of 512 units.
- **GRU:** Similar to the LSTM architecture, but replacing the top hidden layer with a GRU layer of 512 units.
- **RNN:** Similar to the LSTM architecture, but replacing the top hidden layer with a RNN layer of 512 units.

[3] https://github.com/CSTR-Edinburgh/merlin.

# 4 Evaluation and Discussion

In order to achieve our goals and to verify the effectiveness of the proposed method, objective and subjective evaluations were carried out. We conducted two kinds of experimental evaluations. In the first evaluation, we experimentally modeled our continuous vocoder parameters in deep recurrent neural networks by systems given in Sect. 3, and objectively verified. In the second evaluation, we tested them using a subjective listening experiment.

## 4.1 Objective Evaluation

To get an objective picture of how these four RNN systems evaluate against the DNN baseline using the continuous vocoder, the performance of these systems is evaluated by calculating the overall validation error (as mean square error between valid and train values per each iteration) for every training model. The test results for the baseline DNN and the proposed recurrent networks are listed in Table 1. It is confirmed that all parameters generated by the proposed systems presented smaller prediction errors than those generated by the baseline system. More specifically, the BLSTM model can achieve the best results and outperforms other network topologies.

**Table 1.** The objective experimental results for the synthesized speech signal using continuous vocoder

| Systems | Training validation error |
|---|---|
| DNN (baseline) | 1.54 |
| RNN | 1.53 |
| LSTM | 1.53 |
| BLSTM | 1.52 |
| GRU | 1.53 |

## 4.2 Subjective Evaluation

In order to evaluate the perceptual quality of the proposed systems, we conducted a web-based MUSHRA (MUlti-Stimulus test with Hidden Reference and Anchor) listening test [29]. We compared natural sentences with the synthesized sentences from the baseline (DNN), proposed (RNN and BLSTM), and a benchmark system. From the four proposed systems, we only included RNN and BLSTM, because in informal listening we perceived only minor differences between the four variants of the sentences. The benchmark type was the re-synthesis of the sentences with a standard pulse-noise excitation vocoder. In the test, the listeners had to rate the naturalness of each stimulus relative to the reference (which was the natural sentence), from 0 (highly unnatural) to 100 (highly natural). The utterances were presented in a randomized order.

11 participants (7 males, 4 females) with a mean age of 35 years, mostly with engineering background were asked to conduct the online listening test. We evaluated twelve sentences. On average, the test took 11 min to fill. The MUSHRA scores for all the systems are showed in Fig. 3. According to the results, both recurrent networks

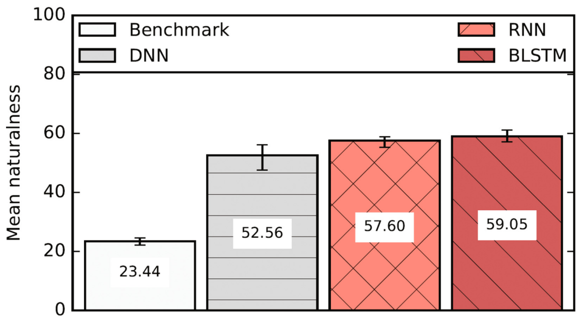

**Fig. 3.** Results of the MUSHRA listening test for the naturalness question. Error bars show the bootstrapped 95% confidence intervals. The score for the reference (natural speech) is not included

outperformed the DNN system (Mann-Whitney-Wilcoxon ranksum test, $p < 0.05$). It is also found that the BLSTM system reached the best naturalness scores in the listening test, consistent with objective errors reported above. However, the difference between RNN and BLSTM is not statistically significant.

## 5 Conclusion

The goal of the work reported in this paper was to apply a Continuous vocoder in recurrent neural network based speech synthesis to enhance the modeling of acoustic features extracted from speech data. We have implemented four deep recurrent architectures: LSTM, BLSTM, GRU, and RNN. Our evaluation focused on the task of sequence modeling which was ignored in the conventional DNN. From both objective and subjective evaluation metrics, experimental results demonstrated that our proposed RNN models can improve the naturalness of the speech synthesized significantly over our DNN baseline. These experimental results showed the potential of the recurrent networks based approaches for SPSS. In particular, the BLSTM network achieves better performance than others.

For future work, the authors plan to investigate other recurrent network architectures to train and refine our continuous parameters. In addition, we will try to implement firstly a mixture density recurrent network and then combining this with BLSTM-RNN based TTS.

**Acknowledgements.** The research was partly supported by the VUK (AAL-2014-1-183) and the EUREKA/DANSPLAT projects. The Titan X GPU used for this research was donated by NVIDIA Corporation. We would like to thank the subjects for participating in the listening test.

## References

1. Zen, H., Tokuda, K., Black, A.: Statistical parameteric speech synthesis. Speech Commun. **51**(11), 1039–1064 (2009)
2. Zen, H., Shannon, M., Byrne, W.: Autoregressive models for statistical parametric speech synthesis. IEEE Trans. Acoust. Speech Lang. Process. **21**(3), 587–597 (2013)
3. Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura T.: Simultaneous modeling of spectrum, pitch, and duration in HMM based speech synthesis. In: Proceedings of Eurospeech, pp. 2347–2350 (1999)
4. Ling, Z.H., et al.: Deep learning for acoustic modeling in parametric speech generation: a systematic review of existing techniques and future trends. IEEE Sig. Process. Mag. **32**(3), 35–52 (2015)
5. Najafabadi, M., Villanustre, F., Khoshgoftaar, T., Seliya, N., Wald, R., Muharemagic, E.: Deep learning applications and challenges in big data analytics. J. Big Data **2**(1), 1–21 (2015)
6. Zen, H., Senior, A., Schuster, M.: Statistical parametric speech synthesis using deep neural networks. In: Proceedings of ICASSP, pp. 7962–7966 (2013)
7. Valentini-Botinhao, C., Wu, Z., and King, S.: Towards minimum perceptual error training for DNN-based speech synthesis. In: Interspeech, pp. 869–873 (2015)
8. Wu, Z., Valentini-Botinhao, C., Watts, O., and King, S.: Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In: ICASSP, pp. 4460–4464 (2015)
9. Zen, H., Senior, A.: Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In: ICASSP, pp. 3844–3848 (2014)
10. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Networks **5**(2), 157–166 (1994)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Fan, Y., Qian Y., Xie F., Soong, F.K.: TTS synthesis with bidirectional LSTM based recurrent neural networks. In: Interspeech, pp. 1964–1968 (2014)
13. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling arXiv preprint: 1412.3555 (2014)
14. Csapó, T.G., Németh, G, and Cernak M.: Residual-based excitation with continuous F0 modeling in HMM-based speech synthesis. In: 3rd International Conference on Statistical Language and Speech Processing, SLSP 2015, vol. 9449, pp. 27–38 (2015)
15. Garner, P.N., Cernak, M., Motlicek, P.: A simple continuous pitch estimation algorithm. IEEE Sig. Process. Lett. **20**(1), 102–105 (2013)
16. Drugman, T., Stylianou, Y.: Maximum voiced frequency estimation: exploiting amplitude and phase spectra. IEEE Sig. Process. Lett. **21**(10), 1230–1234 (2014)
17. Csapó, T.G., Németh, G., Cernak, M., Garner, P.N.: Modeling unvoiced sounds in statistical parametric speech synthesis with a continuous vocoder. In: EUSIPCO, Budapest (2016)
18. Al-Radhi, M.S., Csapó T.G., and Németh, G.: Continuous vocoder in deep neural network based speech synthesis. In: Preparation (2017)
19. Tokuda, K., Kobayashi, T., Masuko, T., Imai, S.: Mel-generalized cepstral analysis – a unified approach to speech spectral estimation. In: Proceedings of ICSLP, pp. 1043–1046 (1994)
20. Imai, S., Sumita, K., Furuichi, C.: Mel log spectrum approximation (MLSA) filter for speech synthesis. Electron. Commun. Jpn. (Part I: Commun.) **66**(2), 10–18 (1983)

21. Al-Radhi, M.S., Csapó, T.G., Németh, G.: Time-domain envelope modulating the noise component of excitation in a continuous residual-based vocoder for statistical parametric speech synthesis. In: Interspeech (2017)
22. Robel, A., Villavicencio, F., Rodet, X.: On cepstral and all-pole based spectral envelope modeling with unknown model order. Pattern Recogn. Lett. **28**(11), 1343–1350 (2007)
23. Galas, T., Rodet, X.: An improved cepstral method for deconvolution of source-filter systems with discrete spectra. In: Proceedings of the ICMC, pp. 82–84 (1990)
24. Cappe, O., Moulines, E.: Regularization techniques for discrete cepstrum estimation. IEEE Sig. Process. **3**(4), 100–103 (1996)
25. Wu, Z., Watts, O., King, S.: Merlin: an open source neural network speech synthesis system. In: Proceedings of the 9th ISCA Speech Synthesis Workshop, Sunnyvale, USA (2016)
26. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. IEEE Trans. on Signal Processing **45**(11), 2673–2681 (1997)
27. Kominek, J., Black, W.: CMU ARCTIC databases for speech synthesis. Language Technologies Institute (2003). http://festvox.org/cmu_arctic/
28. Imai, S., Kobayashi, T., Tokuda, K., Masuko, T., Koishida, K., Sako, S., Zen, H.: Speech signal processing toolkit (SPTK) (2016)
29. ITU-R Recommendation BS.1534. Method for the subjective assessment of intermediate audio quality (2001)