DNN-Based Speech Synthesis: Importance of Input Features and Training Data

Alexandros Lazaridis(✉), Blaise Potard, and Philip N. Garner

Idiap Research Institute, Martigny, Switzerland
{alaza,blaise.potard,phil.garner}@idiap.ch

Abstract. Deep neural networks (DNNs) have been recently introduced in speech synthesis. In this paper, an investigation on the importance of input features and training data on speaker dependent (SD) DNN-based speech synthesis is presented. Various aspects of the training procedure of DNNs are investigated in this work. Additionally, several training sets of different size (i.e., 13.5, 3.6 and 1.5 h of speech) are evaluated.

Keywords: Text-to-speech synthesis · Statistical parametric synthesis · Deep neural networks · Hidden markov models

1 Introduction

Much of the text-to-speech (TTS) work at Idiap is in the context of speech-to-speech translation (S2ST). To this end, good quality speech recognition and synthesis are prerequisites. Further, the translation scenario requires both technologies to exist in multiple languages. Data can be scarce for some languages.

Hidden Markov model (HMM)-based TTS approaches have become dominant in TTS for S2ST, mainly due to their adaptation abilities and flexibility in changing voice characteristics (e.g. speaker, speaking style, emotional state, etc.), using a relatively small amount of data [10], occasionally outperforming even unit-selection approaches [13]. Nonetheless, various limitations and drawbacks occur in HMM-based TTS as listed in the work of Zen et al. [12].

Trying to address some of these deficiencies, deep neural networks (DNNs) have been introduced in speech synthesis over the last few years, outperforming HMM-based TTS approaches, such as Zen et al. [12] using 33k sentences of speech for training and Qian et al. [7] where a more modest database of approximately 5k sentences (approx. 5 h) was used. In an attempt to use an even smaller database, Lu et al. [5], used a training set of 1k sentences in a framework combining Vector Space Representation (VSR) [8] and DNN modelling without managing to outperform the HMM-based one. It seems that there is a data threshold below which HMMs are superior to DNNs.

In the following sections, we describe experiments aimed at evaluating whether recent DNN technology described above could be beneficial in our S2ST scenario. The main focus is on the amount of training data. We hypothesize that although the DNNs will clearly show their superiority over the HMMs when the

A. Ronzhin et al. (Eds.): SPECOM 2015, LNAI 9319, pp. 193–200, 2015.
DOI: 10.1007/978-3-319-23132-7_24

amount of training data is relatively large (i.e. more that 5 or 10 h of speech), this might not be the case when relatively small amount of data (i.e. approximately 1 h of speech) is used. Moreover, other aspects of the training procedure such as model order in terms of layers and nodes per layer, along with positional information in the input layer are also examined.

2 Framework

A DNN is a feed-forward artificial neural network with multiple hidden layers between the input and output layer, creating a mapping function between the input (i.e. linguistic features) vector and the output (i.e. acoustic features) vector. In the training phase, the input text is processed and transformed into labels, which contain linguistic features in an appropriate format for training the DNNs, i.e., containing binary and numerical features. Back-propagation is used for training the DNN using the input and output data.

In the synthesis phase, the input text is processed by the same front-end as in the training phase, creating the input vectors and the trained DNN is used in a forward-propagation manner for mapping them to output vectors. Consequently the acoustic features are created using maximum likelihood parameter generation (MLPG) trajectory smoothing [11] and finally, a vocoder is used for synthesizing the final waveform.

2.1 Database and Input/Output Features

For the experiments the blizzard-challenge-2011 [4] database was used. The speaker is known as "Nancy" and is a US English native female speaker. The database consists of 16.6 h of data, comprising around 12k utterances. The audio was re-sampled to a sampling frequency of 16 kHz for these experiments.

For the training of the DNNs, three different sizes of the *training* set were implemented, i.e. *T13.5*: 13.5 h (10k sentences), *T3.6*: 3.6 h (2.6k sentences) and *T1.5*: 1.5 h (1.1k sentences). The *development* set and the *evaluation* set consisted of 1.35 h (1k sentences) and 0.4 h (0.3k sentences) of speech respectively.

The text corresponding to each audio file has to be converted into a sequence of labels suitable for HMM and DNN training. A conventional and freely available TTS front-end was used for this [1].

The text is turned into a sequence of labels, which contain segmental information and rich contextual parameters such as lexical stress and relative position within syllables, phrases or sentences. The standard "*full*" labels generated by the scripts, i.e. quinphone segmental information, and a large number of categorical, numeric, or binary linguistic and prosodic information, was used [13]. These labels were aligned with the speech signal through a phone-based forced alignment procedure, using the Kaldi toolkit [6]. The models for the alignment were trained on the training plus development sets, and state-level labels force-aligned to acoustic frame boundaries were generated for the training, development and evaluation sets.

Concerning the output features, the STRAIGHT [13] vocoder was used for the acoustic analysis and feature extraction, essentially using the default settings from the EMIME [9] scripts: 5ms sampling step, STRAIGHT Mel-cepstral analysis with 40 coefficients, single f0 value, and 21 coefficients for band aperiodic energy, extracted by the STRAIGHT vocoder. For each acoustic feature, derivatives of first and second order are added. The overall acoustic vector dimension is 186.

2.2 DNN Setup

A slightly modified version of the Kaldi toolkit for the DNN training was used. An automatic procedure was used to convert the labels into numeric values: the categorical data (such as segmental information) was turned into arrays of binary values, while the numerical and binary data was preserved.

Since training requires a frame-level mapping between input labels and acoustic features, the segment-based labels have to be sampled so that we have an input label per acoustic frame. Based on this fact, various implementations were evaluated. Additionally, we hypothesize that the information concerning the input features included in the questions, which are used during the HMM-based TTS training procedure for building the decision tree [13] could be beneficial for training the DNNs. Based on this hypothesis, several sets of binary features were extracted from the questions. The different implementations are the following:

- **DNN**$_{\text{ba}}$: "baseline" DNN system trained using only the states within the phone in the input features, along with the standard "*full*" labels (i.e. a total of 424 input features).
- **DNN**$_{\text{mp}}$: "multi-pos" DNN system trained using the state position within the phone as categorical data, plus using two position features, i.e. numeric values corresponding to the frame position within the current state, and to the frame position within the current segment, plus the standard "*full*" labels (i.e. a total of 431 input features).
- **DNN**$_{\text{mpq}}$: "multi-pos plus phonological questions" DNN system trained using the previous implementation plus some additional phonological information for the current phone extracted based on the questions used in the HMM-based system (i.e. a total of 519 input features).

In some preliminary experiments other implementations of the input feature sets, adding more information extracted based on the question set, were investigated without being beneficial for the DNN-based system after all.

The DNNs were built implementing various combinations of the number of hidden layers (i.e. from 3 to 6 hidden layers), and nodes (i.e. 700, 1000, 1500 and 2000 nodes) in each layer. Each layer comprised an affine component followed by a sigmoid activation function. The input (label) data was further normalized for each component to be of zero mean and unit variance. The output (acoustic) data was normalized globally so that each component had values between 0.01 and 0.99; the output activation function was a sigmoid.

Unlike other approaches (such as Zen [12] or Qian [7]), we did not remove silent frames from the training. The training procedure was standard: we used a stochastic gradient descent based on back propagation. The minimisation criterion was the Mean Square Error (MSE). The training was run on the *training* set, and we used the *development* set for cross-validation.

2.3 HMM Setup

For comparison with state-of-the-art parametric systems, HMM-based synthesis models were built using the HTS v.2.1 toolkit [2]. More specifically, an implementation from the EMIME project [9], freely available online, was employed. We used standard five-state left-to-right Hidden Semi-Markov Models (HSMM), with no-skip.

2.4 Synthesis

The aligned label files from the evaluation set were used for synthesis. In the case of DNN-based synthesis, state-level alignments were used, while in the case of HTS, the alignment was only enforced up to the phone level. In the case of the DNN, synthesis was performed doing a forward pass through the network, followed by acoustic trajectory smoothing [2], through applying the "mlpg" tool from SPTK [3] and global variance computed on each acoustic component. This was followed by resynthesis using the STRAIGHT vocoder. For the HTS system, resynthesis is performed using "HMGenS" with global variance information, followed by STRAIGHT synthesis.

3 Results

In the following subsections, the objective evaluation on the different implementations of the input features and the various sizes of training sets used for training the DNN- and HMM-based systems, as described in the previous section, are presented, along with the subjective evaluation.

3.1 Objective Evaluation

In Fig. 1, the Mel-cepstral distortion (MCD) in dB and the F0 in root mean square error (RMSE) results on the evaluation set, for the different parameters concerning the number of hidden layers (i.e., 3, 4, 5 and 6) and the number of nodes per layer (i.e., 700, 1000, 1500 and 2000), using the full training set (T13.5), for the three DNN-based systems (i.e., DNN_{ba}, DNN_{mp}, DNN_{mpq}) can be seen. In all of the three systems both the MCD error and the RMSE of the F0 are decreasing with the increase of the number of nodes used. Concerning the DNN_{ba} system, the best performance in respect to both MCD error and RMSE, is achieved by the 4-hidden layers and 2000 nodes per layer implementation. The same trend can be seen in the case of the DNN_{mpq} system, while in DNN_{mp}

case, the best performance is achieve by the 5-hidden layers and 2000 nodes per layer implementation. In respect to the three systems, the DNN_{ba} managed to model F0 more accurate followed by the DNN_{mp} one, while in respect to the MCD error, the DNN_{mpq} slightly outperformed and DNN_{mp} one, followed by the DNN_{ba} one.

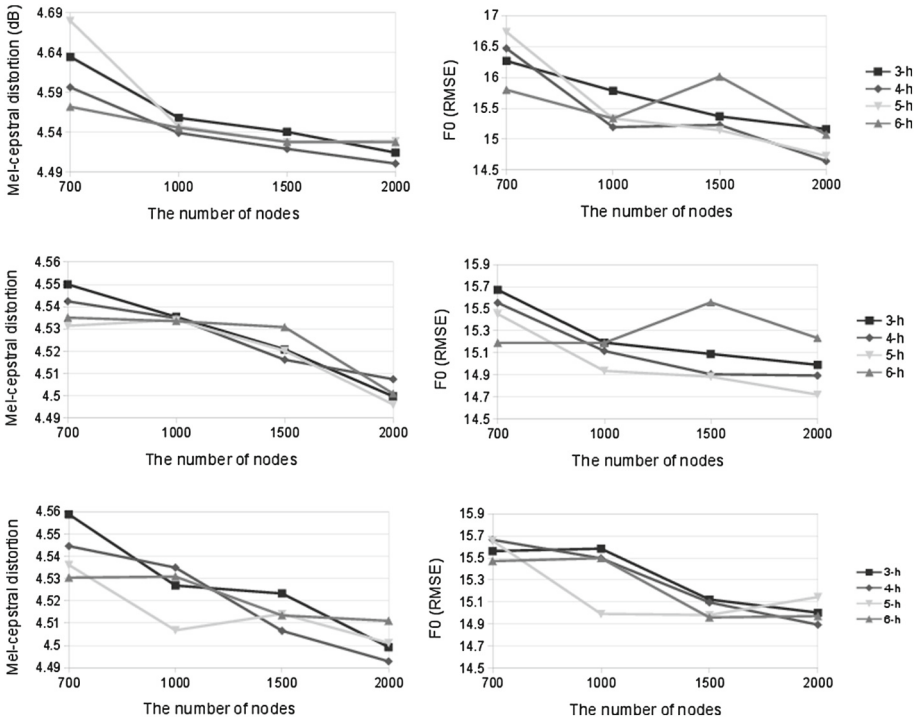

Fig. 1. MCD error in dB and RMSE of F0 in Hz, for the baseline DNN_{ba} (top), multi-pos DNN_{mp} (middle) and multi-pos plus phonological questions features DNN_{mpq} (bottom) systems (trained on T13.5 set), on evaluation set.

In Table 1 the results for the HMM-based system are presented. For comparison reasons, the results for three DNN-based systems are shown again. Along with the MCD error and the RMSE of the F0, the unvoiced/voiced (U/V) error, in percentage, is presented. As can be seen, all DNN-based systems outperform the HMM-based one, achieving around 11 % relative improvement (i.e., reduction) in MCD error, 8–9 % relative improvement (i.e., reduction) in RMSE of the F0 and 30–32 % relative improvement (i.e., reduction) in U/V error.

Based on the aforementioned objective results and an informal subjective one, the DNN_{mp} implementation was chosen for the second part of the experiments. For this, different size of training sets, i.e. the T3.6 training set (3.6 h of speech)

Table 1. MCD in dB, F0 RMSE in Hz and U/V decision errors in %, for the three DNN-based implementations and the HMM-based system (trained on T13.5 set).

System	MCD (dB)	F0 (Hz)	U/V error (%)
HMM	5.052	16.14	9.33
DNN_{ba}	4.501	14.65	6.39
DNN_{mp}	4.496	14.72	6.54
DNN_{mpq}	4.493	14.89	6.38

Table 2. MCD in dB, F0 RMSE in Hz and U/V decision errors in %, for the three different training data sets for the DNN-based and the HMM-based systems.

System	Training set	MCD (dB)	F0 (Hz)	U/V error (%)
HMM	T13.5	5.052	16.14	9.33
DNN_{mp}	T13.5	4.496	14.72	6.54
HMM	T3.6	5.089	16.94	9.17
DNN_{mp}	T3.6	4.563	16.58	6.98
HMM	T1.5	5.166	17.53	10.64
DNN_{mp}	T1.5	4.741	18.72	7.53

and the T1.5 training set (1.5 h of speech), were used for training both HMM-based and DNN-based systems. In some initial experiments, the implementations of 3, 4 and 5 hidden layers with 1000, 1500 and 2000 nodes per layer DNNs, were investigated. The case with the 4 hidden layers with 2000 nodes per layer gave the best performance and are presented next.

In Table 2 the results for the aforementioned experiments can be seen, along with the respective results with T13.5 for comparison reasons. As can be seen, in all cases, the respective DNN-based system outperforms the HMM-based corresponding one in terms of MCD error and U/V error. In respect to RMSE of the F0, only in the case of T1.5, the DNN-based system cannot manage to outperform the HMM-based one.

It should be mentioned here, that various implementations of pre-training were investigated. The deep belief network (DBN) framework [7] was used, trained on different size training sets and with or without splicing features along several frames, but none of them managed to help improve the DNN-based systems compared to the corresponding ones without pre-training.

3.2 Subjective Evaluation

In order to verify the objective evaluation results, a subjective preference listening (ABX) test was conducted. The subjective test was composed of two parts. In the first one, the listeners were asked to state their preference – in terms of the naturalness of the speech – between the DNN- and the HMM-based system,

in respect to the three different training sets (i.e., T13.5, T3.6 and T1.5). In the second part, they were asked to compare each one of the three DNN-based systems (i.e., T13.5, T3.6 and T1.5) in respect to the other two. In the ABX preference test, for each sample, there were 5 preference choices: (1) the first sample sounds much closer to the reference, (2) the first sample sounds a bit closer to the reference, (3) no sample is significantly better than the other, (4) the second sample sounds a bit closer to the reference, (5) the second sample sounds much closer to the reference. Two sets of 7 samples were randomly selected from the evaluation set, and 11 and 19 listeners respectively, participated in the tests. In Table 3 the ABX results concerning the first part of the subjective test, are presented. As can be seen, in all three cases, the DNN-based systems clearly outperform the HMM-based ones.

Table 3. ABX preference test results (%) comparing the DNN-based system with the corresponding HMM-based one for the three different training sets, T13.5, T3.6 and T1.5.

Training set	Strong pref. DNN	Pref. DNN	Equal	Pref. HMM	Strong pref. HMM
T13.5	33.3	37.1	9.5	14.3	5.7
T3.6	20.5	41.9	16.7	17.6	3.3
T1.5	24.8	48.1	14.8	9.5	2.9

Table 4. ABX preference test results (%) comparing the DNN-based systems trained with the three different training sets with each other.

Strong pref. DNN (T13.5)	Pref. DNN (T13.5)	Equal	Pref. DNN (T3.6)	Strong pref. DNN (T3.6)
5.7	20.5	63.3	10.0	0.5
Strong pref. DNN (T3.6)	Pref. DNN (T3.6)	Equal	Pref. DNN (T1.5)	Strong pref. DNN (T1.5)
5.2	25.7	52.9	13.3	2.9
Strong pref. DNN (T13.5)	Pref. DNN (T13.5)	Equal	Pref. DNN (T1.5)	Strong pref. DNN (T1.5)
10.0	40.0	40.5	9.0	0.5

In Table 4 the ABX results concerning the second part of the subjective test, are shown. From these results, it can be seen that there is a preference on the DNN-based system trained using T13.5, over T3.6 and T1.5, nonetheless, in all cases the "no sample is significantly better than the other" choice in the ABX test gathers very high scores (i.e. 40–65 %). These results are in agreement with the objective results, which have shown the clear superiority of the DNN-based systems in respect to the HMM-based ones, even (to our surprise) when a relatively small amount of training data is used, and in parallel the relatively small differences among the three DNN-based systems.

4 Conclusions

Our attempt to explore features extracted from the question set, used in HMM-based techniques, turned out not to be as beneficial as expected. However, even

without the contribution of these features, as both the objective and subjective results clearly show, the DNN-based systems managed to outperform the respective HMM-based ones, even when the smallest training dataset, i.e. 1.5 h of speech, is used for training the systems. Our future focus will be on the adaptation aspects of DNN-based speech synthesis, which is essential in the field of statistical parametric speech synthesis, especially in the area of S2ST.

Acknowledgements. This work has received funding from the Swiss National Science Foundation under the SIWIS project and was supported by Eurostars Programme powered by Eurostars and the European Community under the project "D-Box: A generic dialog box for multi-lingual conversational applications".

References

1. Black, A., Taylor, P., Caley, R.: The festival speech synthesis system: system documentation (1.3.1). Technical report HCRC/TR-83, Human Communication Research Centre (December 1998)
2. HTS: HMM-based speech synthesis system version 2.1 (2010)
3. Imai, S., Kobayashi, T.: Speech signal processing toolkit (SPTK) version 3.7 (2013)
4. King, S., Karaiskos, V.: The Blizzard challenge 2011 (2011)
5. Lu, H., King, S., Watts, O.: Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis. In: SSW8, pp. 281–285 (August 2013)
6. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al.: The Kaldi speech recognition toolkit. In: Proceeding of ASRU (2011)
7. Qian, Y., Fan, Y., Hu, W., Soong, F.: On the training aspects of deep neural network (DNN) for parametric tts synthesis. In: ICASSP, pp. 3829–3833 (2014)
8. Watts, O.: Unsupervised Learning for Text-to-Speech Synthesis. Ph.D. thesis, University of Edinburgh (2012)
9. Wester, M., Dines, J., Gibson, M., Liang, H., Wu, Y.J., Saheer, L., King, S., Oura, K., Garner, P.N., Byrne, W., Guan, Y., Hirsimäki, T., Karhila, R., Kurimo, M., Shannon, M., Shiota, S., Tian, J., Tokuda, K., Yamagishi, J.: Speaker adaptation and the evaluation of speaker similarity in the EMIME speech-to-speech translation project. In: SSW7, pp. 192–197 (2010)
10. Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., Isogai, J.: Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm. Trans. Audio Speech Lang. Proc. **17**(1), 66–83 (2009)
11. Zen, H., Tokuda, K., Kitamura, T.: Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. Comput. Speech Lang. **21**, 153–173 (2006)
12. Zen, H., Senior, A., Schuster, M.: Statistical parametric speech synthesis using deep neural networks. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7962–7966 (2013)
13. Zen, H., Tokuda, K., Black, A.W.: Statistical parametric speech synthesis. Speech Commun. **51**(11), 1039–1064 (2009)