A DEEP RECURRENT APPROACH FOR ACOUSTIC-TO-ARTICULATORY INVERSION

Peng Liu[1,2], *Quanjie Yu*[1,2], *Zhiyong Wu*[1,2,3], *Shiyin Kang*[3], *Helen Meng*[1,3], *Lianhong Cai*[1,2]

[1]Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems,
Shenzhen Key Laboratory of Information Science and Technology,
Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China
[2]Tsinghua National Laboratory for Information Science and Technology (TNList),
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[3]Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China

{liup12, yuqj13}@mails.tsinghua.edu.cn,
{zywu, sykang, hmmeng}@se.cuhk.edu.hk, clh-dcs@tsinghua.edu.cn

ABSTRACT

To solve the acoustic-to-articulatory inversion problem, this paper proposes a deep bidirectional long short term memory recurrent neural network and a deep recurrent mixture density network. The articulatory parameters of the current frame may have correlations with the acoustic features many frames before or after. The traditional pre-designed fixed-length context window may be either insufficient or redundant to cover such correlation information. The advantage of recurrent neural network is that it can learn proper context information on its own without the requirement of externally specifying a context window. Experimental results indicate that recurrent model can produce more accurate predictions for acoustic-to-articulatory inversion than deep neural network having fixed-length context window. Furthermore, the predicted articulatory trajectory curve of recurrent neural network is smooth. Average root mean square error of 0.816 mm on the MNGU0 test set is achieved without any post-filtering, which is state-of-the-art inversion accuracy.

Index Terms— long short term memory (LSTM), recurrent nueral network (RNN), mixture density network (MDN), layer-wise pre-training

1. INTRODUCTION

Human speech is bimodal in nature. There are direct connections between the configurations of the articulators, which are the positions and movements of the lips, tongue, velum, etc., and the acoustic speech. The acoustic-to-articulatory inversion mapping problem involves the inference of the articulator configurations from the acoustic speech signal. There are several potential benefits of acoustic-to-articulatory inversion for speech related applications. In speech recognition, the estimated articulatory parameters or gestures can provide additional speech production knowledge to improve the recognition performance [1]. In speech synthesis, articulatory features can be incorporated into the traditional speech synthesis method to modify the characteristics of the synthesized speech [2]. In computer-aided pronunciation training (CAPT), the recovered vocal tract outlines can be visualized and presented to the users so that they can get a better idea of the articulatory movements in perceptual training [3].

Recently, abundant articulography datasets which include articulatory position data and acoustic data have become available. This promotes the application of machine learning methods such as artificial neural networks [4] or hidden Markov models [5] to tackle the acoustic-to-articulatory inversion problem. A trajectory mixture density network is proposed in [6] to get smooth articulatory parameters. In [7], state-of-the-art inversion accuracies have been obtained by introducing and implementing two deep architectures. But these works are dependent on a predefined fixed-length context window for acoustic input and only use piecewised projections to estimate articulatory configuration, which neglects the temporal correlations in the whole sequence of acoustic features. The inversion results of these works are jagged to some extent. Although using maximum likelihood parameter generation (MLPG) [8] algorithm as a post-process relieves the problem to some extent, it is still important to learn the temporal correlations through sequence in the acoustic-to-articulatory inversion.

Inspired by the success of deep learning architectures in the speech synthesis task [9, 10, 11, 12, 13], we hypothesize that a recurrent architecture would be able to achieve performance improvement in learning the smooth characteristics in the data. We implement a deep bidirectional long short term memory (DBLSTM) recurrent neural network (RNN) and a deep recurrent mixture density network (DRMDN) for the acoustic-to-articulatory inversion task. With these two architectures, the predicted articulatory movement trajectories are quite smooth without the requirement of any post-filtering process. Our methods even perform better than deep trajectory mixture density network [7] and achieve the state-of-the-art inversion accuracies.

2. DEEP BIDIRECTIONAL LSTM RECURRENT NEURAL NETWORK (DBLSTM)

Inserting cyclical connections in a feedforward neural network, we obtain recurrent neural network (RNN). In principle, RNN can map the history of previous input vectors to each output vector. The re-

current connections are able to remember previous inputs and allow them to persist in the networks internal state. So the previous inputs influence the current network output. The feedforward process of RNN is:

$$\mathbf{h}_t = \mathcal{H}(\mathbf{W}_{\mathrm{xh}}\mathbf{x}_t + \mathbf{W}_{\mathrm{hh}}\mathbf{h}_{t-1} + \mathbf{b}_{\mathrm{h}}) \quad (1)$$

$$\mathbf{y}_t = \mathbf{W}_{\mathrm{hy}}\mathbf{h}_t + \mathbf{b}_{\mathrm{y}} \quad (2)$$

$\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_T)$ is a input sequence for time $t = 1 \ ... \ T$, $\mathbf{h} = (\mathbf{h}_1, ..., \mathbf{h}_T)$ is the hidden state sequence computed from $\mathbf{x}$, and $\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_T)$ is the output sequence. $\mathbf{W}_{\mathrm{xh}}$, $\mathbf{W}_{\mathrm{hh}}$ and $\mathbf{W}_{\mathrm{hy}}$ denote the input-hidden, hidden-hidden and hidden-output weight matrices repectively. $\mathbf{b}_{\mathrm{h}}$ and $\mathbf{b}_{\mathrm{y}}$ respectively are hidden and output bias vectors.

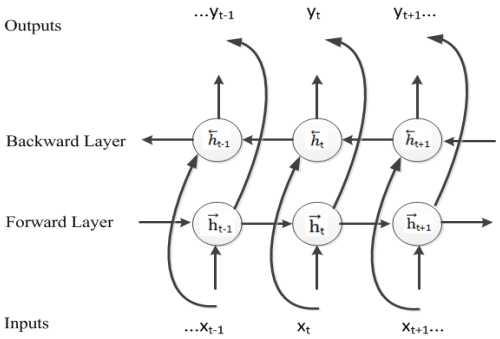

Fig. 1. Bidirectional recurrent neural network (BRNN).

Conventional RNNs can only access previous inputs. However, in articulatory inversion, the movement of the articulators may also have the correlations with the future acoustic features, such as the coarticulation phenomenon. Hence, it is desireable to incorporate the future acoustic context for the articulatory inversion problem. Bidirectional RNN (BRNN) [14], as illustrated in Fig. 1, is able to access past and future context by processing data in both directions. A BRNN computes both forward hidden sequence $\overrightarrow{\mathbf{h}}$, and backward hidden sequence $\overleftarrow{\mathbf{h}}$. The iterative process is:

$$\overrightarrow{\mathbf{h}}_t = \mathcal{H}(\mathbf{W}_{\mathrm{x}\overrightarrow{\mathrm{h}}}\mathbf{x}_t + \mathbf{W}_{\overrightarrow{\mathrm{h}}\overrightarrow{\mathrm{h}}}\overrightarrow{\mathbf{h}}_{t-1} + \mathbf{b}_{\overrightarrow{\mathrm{h}}}) \quad (3)$$

$$\overleftarrow{\mathbf{h}}_t = \mathcal{H}(\mathbf{W}_{\mathrm{x}\overleftarrow{\mathrm{h}}}\mathbf{x}_t + \mathbf{W}_{\overleftarrow{\mathrm{h}}\overleftarrow{\mathrm{h}}}\overleftarrow{\mathbf{h}}_{t-1} + \mathbf{b}_{\overleftarrow{\mathrm{h}}}) \quad (4)$$

$$\mathbf{y}_t = \mathbf{W}_{\overrightarrow{\mathrm{h}}\mathrm{y}}\overrightarrow{\mathbf{h}}_t + \mathbf{W}_{\overleftarrow{\mathrm{h}}\mathrm{y}}\overleftarrow{\mathbf{h}}_t + \mathbf{b}_{\mathrm{y}} \quad (5)$$

In the above equations, $\mathcal{H}$ is usually a sigmoid function. Unfortunately, because of the vanishing gradient problem, RNNs or BRNNs can only access a limited range of context. Long short term memory (LSTM) [15] architecture is proposed to solve the problem. LSTM network uses purpose-built memory cells to exploit long range of context. A single memory cell is illustrated in Fig. 2. For LSTM, $\mathcal{H}$ is implemented by the following functions [16]:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{\mathrm{xi}}\mathbf{x}_t + \mathbf{W}_{\mathrm{hi}}\mathbf{h}_{t-1} + \mathbf{W}_{\mathrm{ci}}\mathbf{c}_{t-1} + \mathbf{b}_{\mathrm{i}}) \quad (6)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{\mathrm{xf}}\mathbf{x}_t + \mathbf{W}_{\mathrm{hf}}\mathbf{h}_{t-1} + \mathbf{W}_{\mathrm{cf}}\mathbf{c}_{t-1} + \mathbf{b}_{\mathrm{f}}) \quad (7)$$

$$\mathbf{c}_t = \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t \tan \mathbf{h}(\mathbf{W}_{\mathrm{xc}}\mathbf{x}_t + \mathbf{W}_{\mathrm{hc}}\mathbf{h}_{t-1} + \mathbf{b}_{\mathrm{c}}) \quad (8)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{\mathrm{xo}}\mathbf{x}_t + \mathbf{W}_{\mathrm{ho}}\mathbf{h}_{t-1} + \mathbf{W}_{\mathrm{co}}\mathbf{c}_{t-1} + \mathbf{b}_{\mathrm{o}}) \quad (9)$$

$$\mathbf{h}_t = \mathbf{o}_t \tan h(\mathbf{c}_t) \quad (10)$$

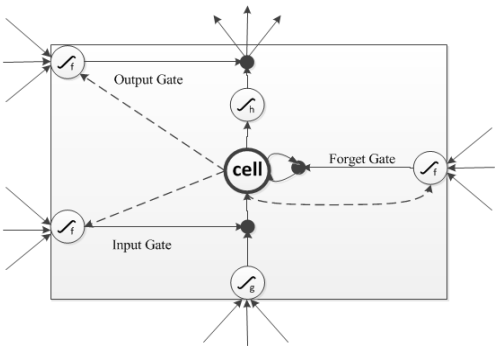

Fig. 2. Long short term memory (LSTM) cell [15].

where σ is a logistic function; i, f, o and c respectively represent input gate, forget gate, outputs gate and cell memory.

Deep bidirectional LSTM (DBLSTM) can be implemented by combining deep BRNN and LSTM. This architecture can learn long range of context.

3. DEEP RECURRENT MIXTURE DENSITY NETWORK (DRMDN)

Articulatory inversion is a challenging task because the same acoustic signal may correspond to different articulatory configurations. [4] used mixture density network (MDN) to represent the uncertainty. The output layer of a MDN specifies a mixture of Gaussians. The MDN takes an input data $\mathbf{x}$ and maps it to a probability distribution over the target domain $\mathbf{t}$. The target probability distribution is a Gaussian mixture model:

$$p(\mathbf{t}|\mathbf{x}) = \sum_{j=1}^{M} \alpha_j(\mathbf{x})\phi_j(\mathbf{t}|\mathbf{x}) \quad (11)$$

where M is the number of mixture components, $\alpha_j(\mathbf{x})$ is the weight

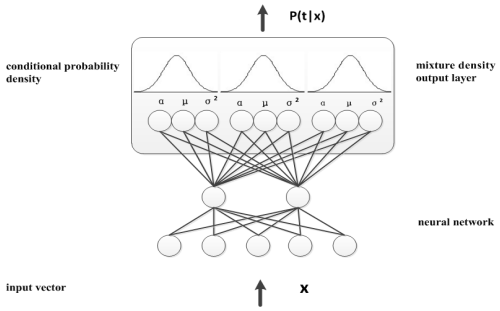

Fig. 3. Mixture density network[17].

of the jth mixture and $\phi_j(\mathbf{t}|\mathbf{x})$ is the probability density of the jth kernel.

A MDN is illustruted in Fig. 3. The network specifies the parameters as follows: $\mathbf{y} = [\mathrm{y}_\alpha^1, \ \mathrm{y}_\mu^1, \ \mathrm{y}_\sigma^1, \ ..., \ \mathrm{y}_\alpha^M, \ \mathrm{y}_\mu^M, \ \mathrm{y}_\sigma^M]$ is the output vector. y_α^j $(j = 1...M)$ go through a softmax and define the weights, $\alpha_j = exp(\mathrm{y}_\alpha^j)/\sum_j exp(\mathrm{y}_\alpha^j)$ and sum up to 1. The means are represented directly by the corresponding outputs, $\mu_j = \mathrm{y}_\mu^j$. The

variance parameters are represented by the corresponding outputs through an exponential function, $\sigma_j = exp(y_\sigma^j)$.

MDN can be trained by minimizing the negative log likelihood of the training data given its parameters. The derivatives with these parameters can be calculated respectively.

A gradient descent method can be used to train the MDN. A deep recurrent mixture density network (DRMDN) can be constructed by replacing the projection output layer of the deep recurrent network by a mixture density output layer. In this approach, the Gaussian mixtures are dependent on a bidirectional recurrent lower structure. We expect that this architecture can learn a more accurate probability distribution based on abundant context information.

4. NETWORK TRAINING

When training the networks, we first conduct layer-wise pre-training. We use rmsprop [18] for its fast converging speed. After pre-training the whole network, we change to stochastic gradient descendent (SGD) which may converge to a little better result at last.

4.1. Incremental layer-wise pre-training

We train the network by an incremental layer-wise method [19], which can further enhance the performance. Besides the input and output layers, the hidden layers are added one by one. The output connections only exist between the current top hidden layer and the output layer. When adding a new layer, we discard the previous output weights and initialize new output weights which connect the new top layer and the output layer; and then all the network weights are tuned. Note that this method is different from the common layer-wise pre-training method where only the weights of the added layer are tuned. This approach assures that each layer has some time in which it is directly connected to the output layer, and then can be effectively trained.

4.2. Rmsprop

Rmsprop is a form of stochastic gradient descent where the gradient is normalized by the magnitude of recent gradients. It has several benefits: 1) it is robust because it utilizes pseudo curvature information, 2) it is applicable of using mini batch learning for it can nicely handle stochastic objectives.

We used a version of rmsprop mentioned in [20]. The weights are updated according to the following equations:

$$n_i = \aleph n_i + (1 - \aleph)\varepsilon_i^2 \tag{12}$$

$$g_i = \aleph g_i + (1 - \aleph)\varepsilon_i \tag{13}$$

$$\Delta_i = \beth \Delta_i - \gimel \frac{\varepsilon_i}{\sqrt{n_i - g_i^2 + \daleth}} \tag{14}$$

$$w_i = w_i + \Delta i \tag{15}$$

We use the following parameters: $\aleph = 0.95, \beth = 0.9, \gimel = 0.0001$, $\daleth = 0.0001$. In the above equations, $\varepsilon_i = \frac{\partial \mathcal{L}(x)}{\partial w_i}$, where w_i is the ith weight. In our experiments, rmsprop converges much faster than the traditional SGD.

5. APPROACHES

We tried 3 systems for articulatory inversion. Layer-wise pre-training and no pre-training experiments are also conducted for each system. All 3 systems and training methods are based on a modified version of RNNLIB [21].

The MNGU0 [22] database is used to evaluate the systems. It contains 1,263 utterances spoken by a single speaker. The database contains parallel electromagnetic midsagittal articulography (EMA) [23] data and line spectral frequencies (LSFs) [24] acoustic data. Each EMA data frame is a 12 dimensional vector. Each dimension corresponds to an x- or y-coordinate of a coil attached in the mid-sagittal plane of the speaker's articulator and there are 6 coils in total. The EMA data are sampled at 200 Hz. The 40 dimentional LSFs together with a gain value are derived from the audio data with 5 msec frame shift to match the sampling rate of the EMA data. The LSF context window is then formed by selecting 10 alternative LSF (and gain value) frames: 5 frames before and 5 frames after the current frame.

5.1. Baseline deep neural networks (DNN)

This system is implemented as the baseline, with which the proposed methods are compared. The static EMA data corresponds to the output of the neural network and a linear projection output layer is added on the top which minimizes the sum of squared error. The LSF context window feeds the network at each time step. Following the configuration setting in [7], we used 4 logistic feedforward layers and each layer has 300 units.

5.2. Deep bidirectional long short term memory (DBLSTM)

In this architecture, we only use the 7th frame of the LSF context window which is 15 msec delay from the current EMA frame. This is the approximate between LSF data and EMA data [25]. The EMA data includes static and dynamic data and we only use the static data.

The first 2 layers are feedforward layers with 300 units. They are supposed to act as a feature extractor. The last 2 layers are bidirectional LSTM layers with 100 units and they are supposed to learn the dynamic property of the training data.

5.3. Deep recurrent mixture density network (DRMDN)

This architecture is formed by replacing the output layer of the previous DBLSTM architecture by a mixture density output layer. We use 4 Gaussian mixtures for the mixture density output layer.

Outputs of the DNN and the DBLSTM are used directly as the predicted articulatory parameters. For DRMDN, we find the mixture with the largest weight and use its mean value as the prediction.

6. EXPERIMENTS AND RESULTS

Two metrics are used to measure the performance of our systems, the root mean-squared error:

$$RMSE = \sqrt{\frac{1}{N}\sum_i (e_i - t_i)^2} \tag{16}$$

where e_i is the network output and t_i is the actual value at time i, and the correlation between the predicted trajectories and actual

articulator trajectories:

$$r = \frac{\sum_i (e_i - \bar{e})(t_i - \bar{t})}{\sqrt{\sum_i (e_i - \bar{e})^2 \sum_i (t_i - \bar{t})^2}} \quad (17)$$

where $\bar{e}$ is the mean value of the predicted value and $\bar{t}$ is the mean of the actual value.

Table 1. *Evaluation of different systems for articulatory inversion.*

Architecture	No pre-training		Layer-wise pre-training	
	RMSE (mm)	r	RMSE (mm)	r
DNN	1.237	0.801	1.000	0.869
DBLSTM	0.963	0.891	0.816	0.921
DRMDN	0.948	0.890	0.832	0.914

We compare the above 3 systems for acoustic-to-articulatory inversion on the MNGU0 database. The results are shown in Table 1. We can see that incremental layer-wise pre-training enhances the performance significantly. However, DRMDN performs not as well as expected. The possible reason will be discussed later in the next section. Fig. 4 illustrates the results of 3 systems with pre-training. The predictions of DNN are jagged trajectories. This is because DNN generates output for each time indepently and is unable to learn proper temporal correlations through the utterance. The predicted trajectories of DBLSTM and DRMDN are almost as smooth as the actural trajectories. We can conclude that DBLSTM and DRMDN use rich context information to generate the current prediction and are able to learn the temporal correlations well.

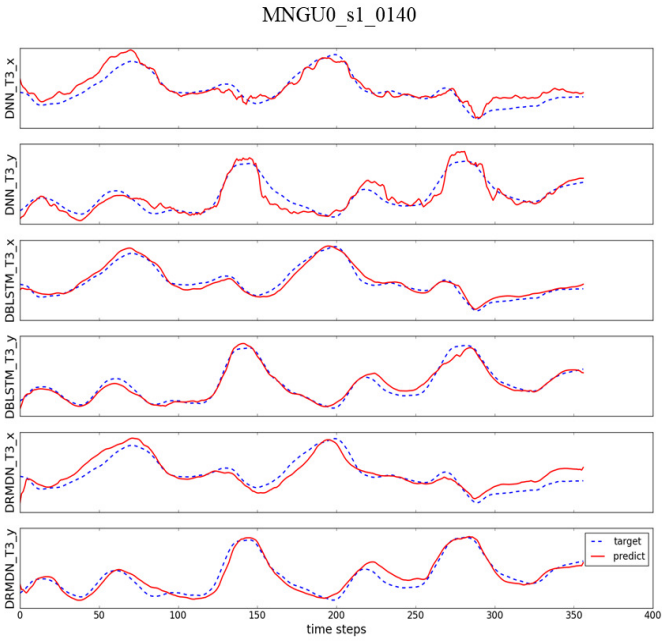

Fig. 4. Samples of the acoustic-to-articulatory inversion results of the 3 systems. T3 is the sensor at tongue dorsum. T3_x and T3_y are respectively the x-, y-coordiante of the T3 sensor.

7. CONCLUSIONS AND DISCUSSIONS

We have implemented two deep recurrent architectures for acoustic-to-articulatory inversion: a deep bidirectional long short term memory network (DBLSTM) and a deep recurrent mixture density network (DRMDN). We suppose that deep recurrent architecture is able to learn the temporal correlations within an utterance. Experiments indicate both of our approaches produce smooth prediction trajectory without any post-filtering.

Both systems perform better than the deep trajectory mixture density network (DTMDN) [7]. Layer-wise pre-training plays an important role in getting good performance. DBLSTM gets the best result which exceeds DTMDN 0.069mm in average RMSE (from 0.885mm to 0.816mm). Bidirectional LSTM layer can learn useful context information inherently. But a fixed-length context window, as used in DTMDN, can only include limited context information within the window. This may explain why our results are better than DTMDN.

However, the DRMDN does not perform better than DBLSTM. [7] also mentioned such phenomenon. A few reasons may explain this: 1) The mean of the most prominent Gaussian can not represent the distribution. Using the mode of mixture of Gaussians may be a better choice. 2) It is difficlut to train a DBLSTM RNN with a Gaussian mixture density output layer. This may remain a future improvement to our current work.

8. ACKNOWLEDGEMENTS

This work is supported by the National Basic Research Program of China (2013CB329304). This work is also partially supported by the National Natural Science Foundation of China (61375027, 61370023 and 61433018) and the Major Project of the National Social Science Foundation of China (13&ZD189). The authors would like to thank Korin Richmond for various advices.

9. REFERENCES

[1] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 121, no. 2, pp. 723–742, 2007.

[2] Z. Ling, K. Richmond, J. Yamagishi, and R. Wang, "Integrating articulatory features into HMM-based parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1171–1185, 2009.

[3] J.H. Zhao, H. Yuan, W.K. Leung, H. Meng, J. Liu, and S.H. Xia, "Audiovisual synthesis of exaggerated speech for corrective feedback in computer-assisted pronunciation training," in *Proc. ICASSP*, 2013, pp. 8218–8222.

[4] K. Richmond, S. King, and P. Taylor, "Modelling the uncertainty in recovering articulation from acoustics," *Computer Speech & Language*, vol. 17, no. 2, pp. 153–172, 2003.

[5] L. Zhang and S. Renals, "Acoustic-articulatory modeling with the trajectory HMM," *IEEE Signal Processing Letters*, vol. 15, pp. 245–248, 2008.

[6] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Proc. INTERSPEECH*, 2006, pp. 577–580.

[7] B. Uria, I. Murray, S. Renals, and K. Richmond, "Deep architectures for articulatory inversion," in *Proc. INTERSPEECH*, 2012, pp. 867–870.

[8] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, 2000, pp. 1315–1318.

[9] S. Kang, X. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Proc. ICASSP*, 2013, pp. 8012–8016.

[10] S. Kang and H. Meng, "Statistical parametric speech synthesis using weighted multi-distribution deep belief network," in *Proc. INTERSPEECH*, 2014, pp. 1959–1963.

[11] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[12] Z. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted boltzmann machines for statistical parametric speech synthesis," in *Proc. ICASSP*, 2013, pp. 7825–7829.

[13] Y. Fan, Y. Qian, F. Xie, and F.K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. INTERSPEECH*, 2014, pp. 1964–1968.

[14] M. Schuster and K.K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, pp. 2673–2681, 1997.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] F.A. Gers, N.N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *The Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2003.

[17] C.M. Bishop, "Mixture density networks," *Technical Report, Aston University*, 1997.

[18] T. Tieleman and G. Hinton, "Lecture 6.5 - rmsprop, coursera: Neural networks for machine learning," 2012.

[19] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2013, pp. 190–198.

[20] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[21] A. Graves, "http://sourceforge.net/projects/rnnl/," .

[22] K. Richmond, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus," in *Proc. INTERSPEECH*, 2009, pp. 1505–1508.

[23] J.S. Perkell, M.H. Cohen, M.A. Svirsky, M.L. Matthies, I. Garabieta, and M.T. Jackson, "Electromagnetic midsagittal articulometer systems for transducing speech articulatory movements," *The Journal of the Acoustical Society of America*, vol. 92, no. 6, pp. 3078–3096, 1992.

[24] H.W. Strube, "Linear prediction on a warped frequency scale," *The Journal of the Acoustical Society of America*, vol. 68, no. 4, pp. 1071–1076, 1980.

[25] C. Qin, M.A. Carreira-Perpinán, and M. Farhadloo, "Adaptation of a tongue shape model by local feature transformations," in *Proc. INTERSPEECH*, 2010, pp. 1596–1599.