Bidirectional LSTM Networks Employing Stacked Bottleneck Features for Expressive Speech-Driven Head Motion Synthesis

Kathrin Haag and Hiroshi Shimodaira

Centre for Speech Technology Research, School of Informatics,
University of Edinburgh, United Kingdom
K.Haag@sms.ed.ac.uk, H.Shimodaira@ed.ac.uk

Abstract. Previous work in speech-driven head motion synthesis is centred around Hidden Markov Model (HMM) based methods and data that does not show a large variability of expressiveness in both speech and motion. When using expressive data, these systems often fail to produce satisfactory results. Recent studies have shown that using deep neural networks (DNNs) results in a better synthesis of head motion, in particular when employing bidirectional long short-term memory (BLSTM). We present a novel approach which makes use of DNNs with stacked bottleneck features combined with a BLSTM architecture to model context and expressive variability. Our proposed DNN architecture outperforms conventional feed-forward DNNs and simple BLSTM networks in an objective evaluation. Results from a subjective evaluation show a significant improvement of the bottleneck architecture over feed-forward DNNs.

Keywords: Head motion synthesis, recurrent neural network, bottleneck feature, long-short-term memory, talking avatar

1 Introduction

Head motion plays an important role in human communication. It is used to give emphasis to certain words or phrases, to convey emotions or to signal agreement or disagreement when listening. In the domain of animation, where realistic virtual agents are desired, it is crucial that head motion looks as natural as possible. Well synthesised head motion can enrich communicative interaction, while badly synthesised head motion is more likely to diminish it.

Work in speech-driven head motion synthesis is often based on Hidden Markov Model (HMM) based methods [1, 2, 3, 4]. In general, frame-wise functions are applied to map acoustic features to head motion angles. In order to compensate for the frame-by-frame independence assumption of HMMs, head motion is classified into typical head motion patterns, either manually or by using automatic clustering. HMMs are then trained on each of these head motion clusters. At synthesis time, for an unknown sequence of acoustic observations, the most likely cluster given the observation has to be recognised first, and then the most

likely head motion sequence is generated from the corresponding HMM that was trained on this cluster.

The data used in these studies typically contains short sentences and/or does not show a large variability of expressiveness in both speech and motion. Constraining the number of possible contexts by pre-defining motion patterns does not work well for expressive data with considerable variation. Furthermore, there is not a one-to-one mapping between speech and head motion [5] and many different output patterns are possible for a given acoustic input sequence. Thus, treating head motion synthesis as a classification problem is not a feasible approach.

DNNs can overcome some of the limitations of the conventional HMM approach. They provide a powerful architecture to capture the large range of variations that are found in expressive data without the need to pre-define motion patterns. Their hidden layers are able to detect complex relationships between input and output features and have been found to be more effective than decision trees [6]. DNNs are also less prone to over-smoothing and preserve more detail in the output signal than HMMs. DNNs have been widely and successfully used in text-to-speech synthesis and often outperform HMM systems [6]. They have also found their way into facial animation [7, 8] and speech-driven gesture synthesis [9].

Ding et al. [10] were the first to use DNNs for speech-driven head motion synthesis. They pre-trained a deep belief network (DBN) with stacked restricted Boltzmann machines, then added a target layer on top of the DBN for parameter fine-tuning. Their training data included broadcast speakers and they used a context window of 11 acoustic frames as input to the DNN. While their architecture performed better than a frame-by-frame DNN modelling approach, a contextual window of this size is not large enough to capture distinctive motion patterns such as nodding and shaking the head, which can span over a window of one or two seconds. For modelling expressive data with a large variability in different motions, an alternative framework is required.

In a further study Ding et al. [11] showed that good performance can be gained by using bidirectional long short-term memory (BLSTM). They report significant improvement of their BLSTM system over a feed-forward DNN, but used data from a single speaker which was not very expressive. We extend on this research and propose a framework which is novel to the domain of head motion synthesis. It combines stacked bottleneck features and a BLSTM network, and we use expressive data.

2 Proposed System

2.1 Bottleneck Features

The features we use for head motion are highly correlated and their dependencies span over long trajectories. The use of bottleneck features for modelling these dependencies seems reasonable. Bottleneck features have been widely used in

speech recognition [15, 16] and text-to-speech systems [17]. They can be used in a similar way for speech-driven head motion synthesis. At first, a DNN with a hidden bottleneck layer is trained on speech and head motion features. This layer has a relatively small number of nodes compared to the other layers in the network. The activations at the bottleneck layer (the bottleneck features) give us a compact frame-wise representation of the input and output features. Multiple bottleneck features of consecutive frames are then stacked using a sliding window and combined with the original speech features as the input to a second DNN network (Fig.1).

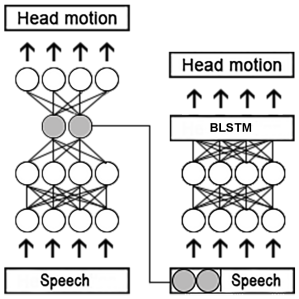

Fig. 1. Example for a bottleneck DNN architecture. Features from the bottleneck layer are stacked and serve as input to a second DNN combined with the original speech features.

2.2 BLSTM with Bottleneck Features

In this paper we investigate whether the use of bottleneck features as input to a BLSTM network is beneficial. It can be argued that contextual information is not required in training a BLSTM network because it already takes the preceding and following context into account, and we will investigate whether this is indeed the case. It should be noted that, although the second network in Fig.1 is trained in a speaker dependent manner to predict an individual speaker's head motion trajectory, the first network can be trained with data from multiple speakers rather than a single one to obtain robust bottleneck features. We found that this results in better prediction of an individual speaker's head motion trajectory. The training procedure is as follows:

1. Train the first DNN which contains a bottleneck layer. The inputs to the DNN are speech features, the output head motion features.
2. Make a forward pass through this network to generate bottleneck features for the training, validation and test data. This is done frame by frame.
3. Stack bottleneck features from the current frame along with n preceding and n following frames.
4. The bottleneck features are combined with the speech features and a second DNN with a BLSTM layer is trained using these features as its input.

5. A forward pass is made through the network to generate head motion features from the second DNN.

2.3 BLSTM Training Issues

When using long segments of input data that go beyond the length of a single sentence, for example when synthesising paragraphs or monologues, we found that BLSTMs are difficult to train and do not generate satisfactory output trajectories. This is especially the case for data with a large expressive variability. One way to work around this is to divide the dataset into smaller segments and employ mini-batch gradient descent. Instead of computing the gradient over all training examples during one iteration, we use a window of w frames, perform one update of the cost function per window and iterate until we reach the end of the data stream. We found that employing mini-batch gradient descent improves the overall performance of our system.

3 Experiments

3.1 Data

We used three male English native speakers from The University of Edinburgh Speaker Personality and MoCap Dataset [18] for training and testing different architectures. This database contains expressive dialogues between semi-professional actors in extroverted and introverted speaking styles. The dialogues were non-scripted and spontaneous. For the purpose of our experiments we selected only the extroverted recordings because they show more variability in head motion and speech.

Speech Features Audio in this database was recorded with a headset microphone at 44.1kHz with 32-bit depth and a MOTU-8pre mixer [19]. Separate recording channels were used for the two speakers and a synchronisation signal was recorded on a third channel in the mixer. For the purpose of this work, the audio signal was down sampled to 16kHz prior to feature extraction. 12 Mel-cepstral coefficients, which represent the discrete log magnitude spectrum, were extracted using SPTK [20]. Voicing probability and energy were computed using openSMILE [21], and smoothed with a moving average filter with a window length of 10 frames.

It has been shown that articulatory features have a closer relationship with head motion than acoustic features [22], even when estimated from speech. Therefore we also extracted articulatory features, which were estimated using an acoustic-to-articulatory inversion technique [22]. They represent (x;y)-coordinates of six active EMA coils (i.e. two coils attached to the upper and lower lip, one to the jaw and three to the tongue). We will refer to them as EMA features. All features were computed from the audio over 25 ms windows at a frame rate of 10 ms to match the frame rate of the head motion data. We also

added their first time derivatives (delta features). The dimension of the speech features was 52.

Head Motion Features The head motion of one speaker of the dialogue pair was recorded with the NaturalPoint Optitrack [23] motion capture system at a 100Hz sampling rate. From the marker coordinates, rotation matrices for head motion were computed using singular value decomposition [24]. The rotation matrices were converted to Euler angles, which describe the motions of pitch, yaw and roll (nodding, shaking and tilting the head). The first and second time derivatives of the Euler angles were also added, resulting in a 9-dimensional vector as the output feature. We used the delta features in training because this resulted in better performance than when only using the static head motion features, but they were not used at synthesis time.

3.2 Preliminary Experiments

We conducted preliminary experiments using data from one speaker to analyse the effects of various hyper-parameters. The results are presented in Fig. 2. We varied the position of the bottleneck layer in order to find its optimal position. Canonical correlations between the original and synthesised head motion were highest when the third layer was set as the bottleneck layer. We also varied the number of nodes in the bottleneck layer. Correlations were highest when using 16 nodes and performance was degraded when using eight or 32 nodes. Thus, we set the nodes in the bottleneck layer to 16. Furthermore, we analysed the effect of the size of the contextual window. We found that highest correlations were achieved when using 20 preceding and 20 following frames.

We also analysed different network topologies for the BLSTM network. An architecture with one or more BLSTM layers and no feed-forward layers resulted in worse performance than when using both feed-forward layers and BLSTM layers. Best performance was achieved with one BLSTM layer on top of two feed-forward layers, which conforms with the findings of text-to-speech synthesis [13, 14]. However, it does not agree with the results of [11] who observed best performance for head motion synthesis when using one BLSTM layers between two feed-forward layers. This suggests that the optimal architecture is dependant on the task and the data being used, and a careful analysis has to be carried out prior to defining the system architecture.

3.3 Experimental Setups

While audio was recorded for both dialogue partners, head motion could only be captured for one speaker. The following architectures use input and output features from this single speaker and include listening pauses (i.e. silences). All systems use the same input and output features. For each speaker we built a speaker-dependent system using four recordings with a duration of approximately four minutes each. Two recordings were used for training, one for validation and one for testing. This was the same for all systems. Training was

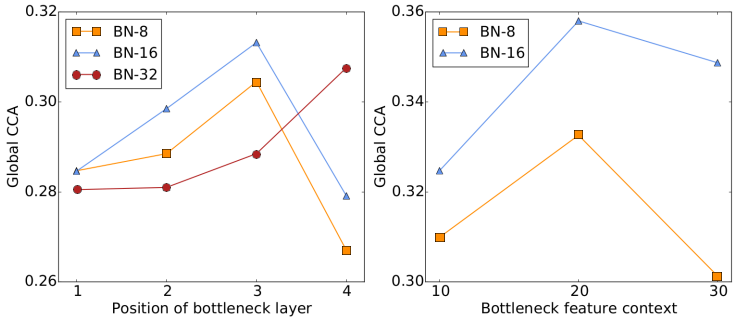

Fig. 2. Analysis of the effect of bottleneck layer position (left) and the number of stacked bottleneck features (right), a number of 10 means that 10 features to the left and 10 features to the right of the current frame were concatenated. Global CCA is defined as the CCA over the entire data stream.

conducted on a GPU using Theano version 0.6. The systems we implemented are summarised as follows:

- **DNN**: This system is our baseline and uses a contextual window of acoustic and EMA features as its input. It is similar to the work of [10] except that we did not use RBMs in pre-training. Acoustic and EMA features were concatenated from a context of five frames to the left and fives frames to the right of the current frame, resulting in a 572-dimensional input vector. We used a conventional feed-forward network employing frame-wise minimum mean squared error criterion and mini-batch in training. The network consisted of three hidden layers with 768 hidden units each. The learning rate was set to 0.002 and halved after 10 epochs, and momentum was 0.3 for the first 10 epochs and increased to 0.9 thereafter. The maximum epoch was 25 and early stopping was applied. A tangent activation function was applied at the bottom layers and a linear output activation function was used.
- **DNN-BN** For this system we used data from all three speakers to generate bottleneck features. The bottleneck features were then used to train a second feed-forward DNN for each of the speakers independently. The bottleneck layer size was set to 16 and we stacked a context of 20 features to the left and 20 features to the right and combined them with our 52-dimensional acoustic and EMA feature vector, resulting in a 708-dimensional input vector. The second DNN had the same architecture as the DNN baseline system and was trained in the same fashion.
- **DNN-BLSTM** For this system we stacked a BLSTM layer on top of two feed-forward layers with tangent activation functions. This system processed the input frame-by-frame using a mini-batch size of 300. The input vector had 52 dimensions and the same hyper-parameters as previously were used.
- **DNN-BLSTM-BN** This system had a similar architecture to the DNN-BN using stacked bottleneck features and acoustic and EMA input features, but the second DNN used a BLSTM layer stacked on top of two feed-forward

layers with tangent activation functions. The second DNN was the same as in DNN-BLSTM.

After generating the output features, the variance of the head motion was re-scaled to match the variance of the head motion in the training data. We applied a least-squares 3-order polynomial smoothing filter on the DNN output.

3.4 Objective Evaluation

We employed canonical correlation analysis (CCA) to measure the correlation between original and synthesised head motion features. Given that $X \in \mathbf{R}^p$ and $Y \in \mathbf{R}^q$ are column vectors with random variables, canonical correlation seeks to find vectors a and b that maximise the correlation τ:

$$\tau = \max_{a,b} corr(a^T X, b^T Y) \tag{1}$$

The advantage of CCA over standard correlation is that CCA can be calculated over multi-column vectors rather than single column vectors. This way we can look at the three Euler angles simultaneously. It is claimed that this procedure finds the highest possible correlation that can be achieved [25].

We define a *local CCA* which computes the canonical correlations over subsets of the data streams [26]. Head motion trajectories change over time and linear correlations rarely hold over the whole data. Therefore it is useful for us to measure the similarity of the original and synthesised head motion using a smaller time window of n frames that starts at t^{th} frame such that

$$r_t = \frac{1}{d}\left(\sum_{i=1}^{d} \text{corr}\left(A^{[i]T} X_{[t:t+n-1]}, B^{[i]T} Y_{[t:t+n-1]}\right)\right) \tag{2}$$

where $A^{[i]}, B^{[i]}$ are the canonical coefficients obtained in the global CCA and d the dimension of features. For local CCA, we used a time window of 300 frames and calculated the average from the resulting scores.

Results The highest local CCA was achieved for DNN-BLSTM-BN while the DNN baseline performed worst. DNN-BLSTM is slightly better than DNN-BN and comes second best. These results suggest that combing stacked bottleneck features and a BLSTM architecture works best, however the difference to the remaining systems is only subtle.

3.5 Subjective Evaluation

A mean opinion score (MOS) test was carried out to evaluate the naturalness of the head motion generated by the four presented systems. Head motion was mapped onto a talking head using the Poser Pro 2012 [27] animation software. Audio was provided as a reference but we refrained from using lip-sync to make

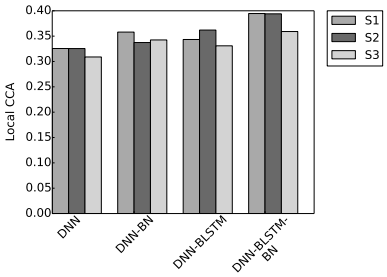

Fig. 3. Average local CCA for the four built systems by speaker before smoothing was applied. *S* refers to the relevant speaker.

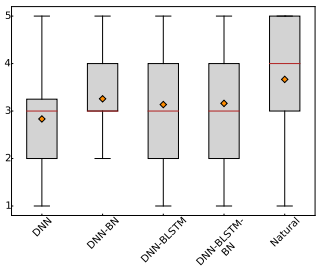

Fig. 4. MOS results - horizontal line indicates the median, diamond shape the mean

the subjects focus only on the head motion. For each system, 16 videos between 8-12 seconds long were animated and four videos with natural speech were added for sanity checking of the ratings. A Latin Square design with four groups was used so that subjects did not watch an animation with the same audio more than once. The subjects were asked to rate the naturalness of the animated head motion on a scale from 1 (very unnatural) to 5 (very natural). 20 English native speakers took part in the test, four of them were excluded in the analysis due to poor naturalness ratings of the natural head motion. Each system was rated 64 times.

Results A MOS was calculated for each system by subject, results are shown in Fig. 4. Listeners were conservative in their judgement of natural head motion, but it was still considered the most natural. We assume that subjects treat the 5-point MOS scale as an interval rather than an ordinal scale [28], thus we applied a one-way ANOVA to compare the means instead of the medians.

The DNN-BLSTM-BN system was only marginally considered as more natural than DNN-BLSTM, but the difference is not significant. DNN-BN seemed to be regarded as slightly more natural than the combined DNN-BLSTM-BN and the DNN-BLSTM system, but no significance can be reported. The only significant difference is between the DNN baseline and DNN-BN ($F=11.5$, $p<0.05$).

4 Conclusions and Further Work

In this paper we proposed using stacked bottleneck features and BLSTMs for expressive head motion synthesis. Our objective evaluation suggests that combining bottleneck features with a BLSTM network outperforms systems that make use of either stacked bottleneck features or BLSTMs. It would also appear that BLSTMs generally work better than feed-forward architectures. However, our subjective evaluation does not confirm this; all contextual systems are on a similar level when rated by subjects. It should be noted that we used a challenging dataset with expressive speech and head motion, better results might be achieved using data with less variation in expressiveness.

References

1. Sargin, M.E., Aran, O., Karpov, A., Ofli, F., Yasinnik, Y., Wilson, S.: Combined Gesture-Speech Analysis and Speech Driven Gesture Sythesis. In: IEEE International Conference on Multimedia and Expo, pp. 893-896, (2006)
2. Busso, C., Deng, Z., Grimm, M., Neumann, U., Narayanan, S.: Rigid Head Motion in Expressive Speech Animation: Analysis and Synthesis. In: IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, pp. 1075-2007, (2007)
3. Ben Youssef, A., Shimodaira, H., Braude, D.A.: Articulatory Features for Speech-Driven Head Motion Synthesis. In: Interspeech 2013, 14th Annual Conference of the International Speech Communication Association, pp. 2758-2762, (2013)
4. Braude, D.A., Shimodaira, H., Ben Youssef, A.: Template-Warping Based Speech Driven Head Motion Synthesis. In: Interspeech 2013, 14th Annual Conference of the International Speech Communication Association, pp. 2763-2767, (2013)
5. Yehia, H.C., Kuratate, T., Vatikiotis-Bateson, E.: Linking Facial Animation, Head Motion and Speech Acoustics. In: Journal of Phonetics, vol. 30, no. 3, pp. 555-568, (2002)
6. 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 (ICASSP), vol. 20, pp. 1713-1724, (2013)
7. Zhao, K., Wu, Z., Cai, L.: A Real-Time Speech Driven Talking Avatar Based on Deep Neural Network. In: Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp. 1-4, (2013)
8. Susskind, J., Hinton, G., Movellan, J., Anderson, A.: Generating Facial Expressions with Deep Belief Nets. In: Or, J. (ed.) Affective Computing, Focus on Emotion Expression, Synthesis and Recognition. I-TECH Education and Publishing, (2008)
9. Chiu, C.-C., Marsella, S.: How to Train your Avatar: A Data-Driven Approach to Gesture Generation. In: Intelligent Virtual Agents, pp. 127-140, Springer, (2011)
10. Ding, C., Xie, L., Zhu, P.: Head Motion Synthesis from Speech Using Deep Neural Networks. In: Multimedia Tools and Applications, vol. 74, pp. 9871-9888, (2014)
11. Ding, C., Zhu, P., Xie, L.: BLSTM Neural Networks for Speech Driven Head Motion Synthesis. In: Interspeech 2015, 16th Annual Conference of the International Speech Communication Association, pp. 3345-3349, (2015)
12. Hochreiter, S.: Recurrent Neural Net Learning and Vanishing Gradient. In: International Journal Of Uncertainity, Fuzziness and Knowledge-Based Systems, vol. 6, no. 2, pp. 107-116, (1998)
13. Fan, Y., Qian, Y., Xie, F.-L., Soong, F.K.: TTS Synthesis with Bidirectional LSTM Based Recurrent Neural Networks. In: Interspeech 2014, 15th

Annual Conference of the International Speech Communication Association, pp. 1964-1968, (2014)
14. Fan, B., Wang, L., Soong, F.K., Xie, L.: Photo-Real Talking Head with Deep Bidirectional LSTM. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4884-4888, (2015)
15. Dong,Y., Seltzer, M.L.: Improved Bottleneck Features Using Pre-Trained Deep Neural Networks. In: Interspeech 2011, 12th Annual Conference of the International Speech Communication Association, pp. 237-240, (2011)
16. Gehring, J., Miao, Y., Metze, F., Waibel, A.: Extracting Deep Bottleneck Features Using Stacked Autoencoders. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3377-3381, (2013)
17. Wu, Z., Valentini-Botinhao, C., Watts, O., King, S.: Deep Neural Networks Employing Multi-Task Learning and Stacked Bottleneck Features for Speech Synthesis. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4460-4464, (2015)
18. Haag, K., Shimodaira, H.: The University of Edinburgh Speaker Personality and MoCap Dataset. In: Proceedings of Facial Analysis and Animation, ACM, pp. 8:1-8:2, (2015)
19. Motu, http://motu.com
20. Speech Signal Processing Toolkit (SPTK), http://sptk.sourceforge.net
21. Eyben, F., Woellmer, M., Schuller, B.: openSMILE: The Munich Versatile and Fast Open-Source Audio Feature Extractor. In: MM '10 Proceedings of the 18th ACM International Conference on Multimedia, ACM, pp. 1459-1462, (2010)
22. Ben Youssef, A., Shimodaira, H., Braude, D.A.: Speech Driven Talking Head from Estimated Articulatory Features. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4606-4610, (2014)
23. NaturalPoint Optitrack, http://www.naturalpoint.com/optitrack
24. Soederkvist, I., Wedin, P.-A.: Determining the Movements of the Skeleton Using Well-Configured Markers. In: Journal of Biomechanics, vol. 26, pp. 1473-1477, (1993)
25. Alpert, M., Peterson, R.: On the Interpretation of Canonical Correlation Analysis. Journal of Marketing Research, vol. 9, pp. 187-192, (1972)
26. Braude, D.: Head Motion Synthesis: Evaluation and a Template Motion Approach. Ph.D. dissertation, University of Edinburgh, School of Informatics, (2016)
27. Poser Pro 2012, http://my.smithmicro.com/poser-3d-animation-software.html
28. Dall, R., Yamagishi, J., King, S.: Rating Naturalness in Speech Synthesis: The Effect of Style and Expectation. In: Proceedings of the 7th International Conference on Speech Prosody, pp. 1012-1016, (2014)