Speech-Driven Head Motion Synthesis Using Neural Networks

Chuang Ding[1], *Pengcheng Zhu*[2], *Lei Xie*[1,2], *Dongmei Jiang*[1], *Zhonghua Fu*[1]

[1]School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]School of Software and Microelectronics, Northwestern Polytechnical University, Xi'an, China

dingchuangnwpu@gmail.com, nwpuzpc@gmail.com, lxie@nwpu.edu.cn

Abstract

This paper presents a neural network approach for speech-driven head motion synthesis, which can automatically predict a speaker's head movement from his/her speech. Specifically, we realize speech-to-head-motion mapping by learning a multi-layer perceptron from audio-visual broadcast news data. First, we show that a generatively pre-trained neural network significantly outperforms a randomly initialized network and the hidden Markov model (HMM) approach. Second, we demonstrate that the feature combination of log Mel-scale filter-bank (FBank), energy and fundamental frequency (F0) performs best in head motion prediction. Third, we discover that using long context acoustic information can further improve the performance. Finally, extra unlabeled training data used in the pre-training stage can achieve more performance gain. The proposed speech-driven head motion synthesis approach increases the CCA from 0.299 (the HMM approach) to 0.565 and it can be effectively used in expressive talking avatar animation.

Index Terms: head motion synthesis, neural network, deep neural network, talking avatar

1. Introduction

We naturally move our heads when we are speaking. Previous study has shown that this rhythmic head motion not only reflects paralinguistic information like stress, intonation, and emotion but also conveys important linguistic information [1]. Since head movements are tied so closely to the prosody of speech, they are also called *visual prosody* [2]. Therefore, besides lip articulation, natural head motion is important to realistic facial animation and engaging human-computer interactions for a lifelike talking avatar.

Some researchers have studied the connection between speech and head motion. Busso *et al.* [3] reported high correlation between head movements and acoustic features via canonical correlation analysis (CCA) [4, 5]. Munhall *et al.* [1] suggested that head motion was an essential ingredient in speech perception and appropriate head motion significantly enhanced the perception of an animated character. Based on these discoveries, researchers have been interested in automatically predicting head motions from either text [6, 7] or speech [5, 8]. Speech-driven head motion synthesis can be addressed as a *classification/recognition* or a *regression* task. The former approaches categorize head motion into several patterns manually or automatically. For example, Graf *et al.* [9] manually categorized head motion into three patterns: nod around one axis, nod with overshoot and abrupt swing in one direction; Busso *et al.* [3] partitioned continuous head motion trajectories into several patterns automatically using LBG-VQ. A statistical framework, e.g. HMM-GMM, was often adopted for speech-driven head motion synthesis, where Gaussian mixture model (GMM) was used to model the correspondences between speech and the head motion patterns while hidden Markov model (HMM) was used to model the time evolution of the head motion patterns. In the synthesis phase, the acoustic input was decoded into a sequence of head motion patterns and a head rotation trajectory was subsequently generated by simply connecting the Gaussian means of the head motion patterns [10]. Inspired by the HMM-based speech synthesis, smooth head motion trajectories can be further generated using dynamic acoustic features [11]. Recent studies have shown that improved performance can be achieved if articulatory features and a better head motion clustering algorithm are used [12, 13].

The performance of classification/recognition methods heavily relies on the definitions of typical head motion patterns [14] and the accurate recognition of these patterns. In addition, the association between speech and head motion is essentially a non-deterministic, many-to-many mapping. As a result, the head motion recognition accuracy remains very low [10]. Therefore, a natural thought is to regard speech-to-head-motion mapping directly as a regression problem. An early attempt was performed by Yehia *et al.* [15], in which a linear estimator (affine transformation) was adopted to map fundamental frequency (F0) to head motion. However, the correlation between the estimated head motion and the real one was considerably low. Recently, instead of using a linear mapping, we used a neutral network (NN) to seek a non-linear mapping from acoustic speech to head motion. Compared with previous approaches, we showed that a simple one-hidden-layer MLP (multi-layer perceptron) with MFCC input improved the head motion prediction accuracy significantly [16].

In the past several years, we have seen a tremendous development of neural networks with effective training strategies and deep architectures [17, 18, 19]. Recently, Uria *et al.* [20] showed that deep architectures and unsupervised pre-training were effective in articulatory inversion, a regression task, in which continuous articulatory movements were accurately predicted from acoustic speech. Compared with the traditional neural network, the success mainly lies in the many-layer architecture and how those layers are optimized. Each layer in the network nonlinearly transforms its input representation into a higher level, resulting in more abstract representation that better models the underlying factors of the data. Therefore, lower representations of the input, e.g., pixels in images [21] and filter banks (FBank) in speech [22], can be effectively used to further boost the performance.

Motivated by the recent development of neural networks, this paper studies how to optimize the neural network approach for speech-driven head motion synthesis. Firstly, we investigate the effectiveness of generative pre-training and the best architecture of a neural network for the speech-to-head-motion

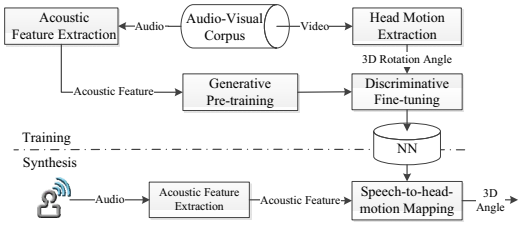

Figure 1: Architecture of the system

synthesis task. Secondly, we evaluate the effectiveness of different acoustic features, including Mel frequency cepstral coefficients (MFCC), log Mel-scale filter-bank (FBank) and prosodic features, i.e., fundamental frequency (F0) and energy. Thirdly, we investigate the head motion prediction ability of different context of the acoustic speech input. Finally, we examine whether extra training data is beneficial to the head motion prediction performance in the unsupervised pre-training stage. Some important conclusions are drawn from our study.

2. System Overview

Figure 1 illustrates the block diagram of the speech-driven head motion synthesis approach that consists of a training phase and a synthesis phase. In the first step of the training phase, acoustic features and head motions are extracted through the acoustic feature extraction and the head motion extraction modules, respectively. Head rotation angles, i.e., nod, yaw and roll are extracted. Subsequently, we train a neural network through a pre-training step and a fine-tuning step. In order to pre-train the NN, we train a series of Restricted Boltzmann Machines (RBMs) using acoustic features and stack up these RBMs to form a dynamic belief network (DBN) architecture. The RBM pertaining procedure is used to initialize the weights of a NN. Then we use back propagation (BP) algorithm to discriminatively fine-tune the NN, which builds up the mapping between acoustic features and the head motion. The synthesis phase is quite simple. Given the acoustic features extracted from a new speech waveform, the three head rotation angles are estimated from the non-linear mappings that the NN embodies.

3. Head Motion Synthesis with NN

Our previous work has shown that a one-hidden layer MLP with random initialization achieves considerable performance in head motion synthesis [16]. This naturally motivates us to use a neural network with more powerful architecture and effective learning method to further push forward the performance of speech-driven head motion synthesis.

3.1. Neural Networks

Multi-layer perceptron (MLP) is a typical neural network with a feed-forward mechanism that maps sets of input data onto a set of outputs. In our case, the input and output are acoustic features and head rotation angles, respectively. An MLP usually consists of an input layer, a hidden layer and an output layer, and the nodes in each layer are fully connected to the nodes in another layer. An MLP with multiple hidden layers is also known as deep neural network (DNN), as shown in Figure 2. The input layer has no computation capability as it simply attaches the observations to the network. Each hidden layer takes in the activations of the layer below and computes a new set of nonlinear activations for the layers above. The

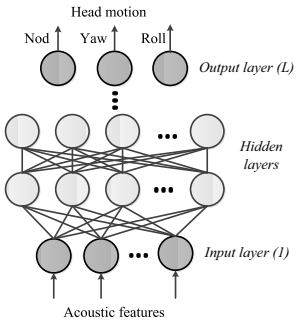

Figure 2: The structure of a neural network

output layer generates either a value in regression or a posterior vector in classification using the activations from the last hidden layer. Each hidden layer computes the activation $\mathbf{h}_l$ via a linear transformation using a weight matrix $\mathbf{W}_l$ and a bias vector $\mathbf{b}_l$ followed by a nonlinear function $f_l(\mathbf{x})$:

$$\mathbf{h}_l = f_l(\mathbf{W}_l\mathbf{h}_{l-1} + \mathbf{b}_l) \quad for \quad 1 \leq l < L \tag{1}$$

where the nonlinear function $f_l(\mathbf{x})$ usually operates in an element-wise manner on the input vector.

The commonly used activation function is the sigmoid logistic function. Each sigmoid hidden unit can be regarded as carrying out a logistic linear regression feature extraction process which refines the input representation to a better one. The output layer (Lth layer) acts as the functional role that is to predict either a value or a class label. In our study, head rotation angles are the targets to be predicted. The output layer simply carries out a similar linear regression as hidden layers do using a weight matrix $\mathbf{W}_L$ and a bias vector $\mathbf{b}_L$. However, a different task-dependent nonlinear function is usually adopted. For regression tasks like our speech-to-head-motion mapping, a linear or sigmoid function is often used. In summary, the parameters for an L-layer network are $(\mathbf{W}_1,\ \mathbf{b}_1)$, $(\mathbf{W}_2,\ \mathbf{b}_2)$, ..., $(\mathbf{W}_L,\ \mathbf{b}_L)$. They are usually randomly initialized and then discriminatively updated using the BP algorithm [23]. However, the gradient diminishing problem usually leads the BP algorithm to local optima [24].

3.2. Deep Belief Network based Pre-training

Hinton *et al.* have proposed a training method by using deep belief networks (DBN) [25], which provides a practical way of building layered networks and triggered great interest in learning deep models. The key of learning deep models lies in the unsupervised generative pre-training using Restricted Boltzmann Machines (RBMs) and stacking RBMs to form a DBN. This generative pre-training stage leads the model into a space that is close to a better optimum. It hence enables the learning of models with better generalizations.

An RBM can be considered as a special type of Markov Random Field (MRF) that has one layer of (typically Bernoulli) stochastic hidden units and one layer of (typically Bernoulli or Gaussian) stochastic visible or observable units. It can be viewed as a bipartite graph in which visible units that represent observations are connected to binary, stochastic hidden units using undirected weighted connections. There are no visible-visible or hidden-hidden connections and all visible units are connected to all hidden units. RBMs have an efficient training procedure which makes them suitable as building blocks for DBN [17].

In an RBM, the joint distribution $p(\mathbf{v}, \mathbf{h};\theta)$ over the visible units $\mathbf{v}$ and hidden units $\mathbf{h}$, given the model parameters θ, is

defined by an energy function $E(\mathbf{v}, \mathbf{h};\theta)$, as depicted in

$$p(\mathbf{v}, \mathbf{h};\theta) = \frac{exp(-E(\mathbf{v}, \mathbf{h};\theta))}{Z}, \quad (2)$$

where

$$Z = \sum_{\mathbf{v}} \sum_{\mathbf{h}} exp(-E(\mathbf{v}, \mathbf{h};\theta)) \quad (3)$$

is a normalization factor and the marginal probability that the model assigns to a visible vector $\mathbf{v}$ is

$$p(\mathbf{v};\theta) = \frac{\sum_{\mathbf{h}} exp(-E(\mathbf{v}, \mathbf{h};\theta))}{Z} \quad (4)$$

For a Bernoulli-Bernoulli RBM, the energy function is

$$E(\mathbf{v}, \mathbf{h};\theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij}v_j h_j - \sum_{i=1}^{I} b_i v_i - \sum_{j=1}^{J} a_j h_j \quad (5)$$

where w_{ij} represents the symmetric interaction term between visible unit v_i and hidden unit h_j, b_i and a_j the bias terms, and I and J are the numbers of visible and hidden units, respectively. The conditional probabilities can be efficiently calculated as

$$p(h_j = 1|\mathbf{v};\theta) = \sigma(\sum_{i=1}^{I} w_{ij}v_i + a_j) \quad (6)$$

$$p(v_i = 1|\mathbf{h};\theta) = \sigma(\sum_{j=1} w_{ij}h_j + b_i) \quad (7)$$

where $\sigma(x) = \frac{1}{1+exp(x)}$. Similarly, for a Gaussian-Bernoulli RBM, the energy is

$$E(\mathbf{v}, \mathbf{h};\theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij}v_i h_j - \frac{1}{2}\sum_{i=1}^{I}(v_i - b_i)^2 - \sum_{j=1}^{J} a_j h_j. \quad (8)$$

The corresponding conditional probabilities become

$$p(h_j = 1|\mathbf{v},\theta) = \sigma(\sum_{i=1}^{I} w_{ij}v_i + a_j) \quad (9)$$

$$p(v_i|\mathbf{h};\theta) = \mathcal{N}(\sum_{j=1}^{J} w_{ij}h_j + b_i, 1) \quad (10)$$

where v_i takes real values and follows a Gaussian distribution with mean $\sum_{j=1}^{J} w_{ij}h_j + b_i$ and variance one. Gaussian-Bernoulli RBMs can be used to convert real-valued stochastic variables to binary stochastic variables, which can then be further processed using the Bernoulli-Bernoulli RBMs [26].

Stacking a number of RBMs learned layer by layer from bottom up gives rise to a DBN. The stacking procedure is as follows [21]. After learning a Gaussian-Bernoulli RBM (used in our study) or Bernoulli-Bernoulli RBM, we treat the activation probabilities of its hidden units as the data for training the Bernoulli-Bernoulli RBM one layer up. The activation probabilities of the second-layer Bernoulli-Bernoulli RBM are then used as the visible data input for the third-layer Bernoulli-Bernoulli RBM. In this way, we grow the network to a desired depth. The greedy procedure above achieves approximate maximum likelihood learning. Note that this learning procedure is unsupervised and requires no target label, in our case, the head motions.

A randomly-initialized target layer is added on the top of the DBN, resulting in a (deep) neural network [17]. In our approach, a linear regression layer is used and the output of this layer corresponds to head rotation angles. The DBN-based pre-training procedure is used to initialize other layers of the network. The standard back propagation algorithm is used to fine-tune the whole NN model.

Table 1: Audio-visual corpus for head motion synthesis

Source	AV data of an anchor from NBC English BN
Training data	93 minutes with head rotation information 73 minutes without head rotation information
Testing data	10 minutes with head rotation information
Acoustic feature	MFCC, FBank, Energy and F0

4. Corpus

We collect 176-minute audio-visual data of a male anchorperson from NBC English broadcast news (BN). We aim to predict the head movement of the anchorperson from his news-reporting speech. The data assignment is summarized in Table 1. The video frame rate is 25fps and the audio sample rate is 44.1KHz. Audio is further down-sampled to 16KHz for acoustic feature extraction. We extract MFCC, FBank, energy features using HTK [27] and F0 features using STRAIGHT [28]. During feature extraction, the frame window length is set to 25ms with an overlap of 15ms. In the experiments, a context window of N acoustic frames is used as the neural network input, where N is tuned for best performance. Hence, the number of the units of the input layer is equal to $D \times N$, where D denotes the dimension of the frame-level acoustic feature vector. We use IntraFace [29] to track the anchor face from the video clips and get the head rotation Euler angles. Both the acoustic features and the head motion angles are processed by utterance-level normalization that subtracts their respective global mean and divides by 4 times the standard deviation for each dimension.

5. Experimental Results

We perform a series of experiments on the audio-visual corpus described in Section 4 and the commonly-used correlation-based criterion, e.g., CCA [4, 10], is adopted for evaluation. For comparison, we also build a baseline HMM-GMM system using HTS [30]. We follow the system setup described in [10] and the only difference is that, instead of manually categorizing the head motion patterns, we use K-means clustering to automatically categorize head motion into 8 patterns. The head pattern recognition accuracy is 47.03% for the test set.

5.1. Result Comparison

Table 2 shows the head motion prediction performances achieved by the HMM-GMM approach [10, 11] and the neural network approach with different hidden layers and initialization strategies. As a sanity check, the performance of randomly generated head motion is also reported (named Random).

The neural networks are trained using 39-D MFCC features (static plus first and second order delta features). The NN input layer has 429 visible units as we use a context span of 11 contiguous frames (5-1-5). The output layer has 3 units, corresponding to the head rotation angles. The network is randomly initialized or generatively pre-trained. For each NN with different hidden layers, we tune the number of hidden units in order to achieve the best performance. All NNs are trained for 50 epochs with a momentum of 0.9. The learning rate is scaled by a factor of 0.99 each epoch. BP is performed using stochastic gradient descent (SGD) in mini-batches of 128 examples.

Table 2 shows that all the neural networks, either randomly initialized or generatively pretrianed, outperform the HMM-GMM approach considerably. The best performance is achieved by the 1-hidden-layer DBN-pretrianed NN that

Table 2: Head motion prediction performance

System	# Layer	CCA
DBN-Pretrained NN	1	**0.511**
	2	0.458
	3	0.382
Random-Initialized NN	1	0.372
	2	0.346
	3	0.312
HMM-GMM	-	0.299
Random	-	0.094

Table 3: The performance of different acoustic features

Feature	MFCC	E_F0	FBank	FBank_E_F0
CCA	0.511	0.323	0.539	**0.554**

improves CCA from 0.299 (HMM-GMM) to 0.511. This indicates that regrading the speech-to-head-motion mapping as a non-linear regression task is more appropriate as compared with a classification task. Another important observation is that generative pre-training can significantly improve performance: DBN-pretrained NNs outperform randomly-initialized NNs by a large margin. We also notice that the best CCA is achieved by a 1-layer network and further increase of network depth can not bring performance gain. This may be explained by two reasons: (1) the relationship between acoustic speech and the head motion is probably a one-layer nonlinear mapping; (2) the size of our training data is limited and the network is overfitted.

5.2. Performance of Different Acoustic Features

We compare different acoustic features to show their abilities in predicting head motions. Specifically, 39-D MFCC, 26-D log Mel-scale filterbank (FBank), 3-D F0, 3-D short-term energy (E) [1] and some feature combinations are investigated. Since head movements are tied so closely to the prosody of speech, we also include the energy and F0 information for head motion prediction. All networks are DBN-pretrained with one hidden layer and a context span of 11 contiguous frames is adopted as input. For each feature configuration, the number of hidden layer units is tuned for best performance. The performance of different acoustic features is shown in Table 3. From the results, we can see that FBank shows superior performance compared with MFCC. This conclusion is consistent with DNN-based speech recognition where FBank also outperforms MFCC [22]. The simple prosodic feature combination E_F0 can achieve a CCA of 0.323, which still outperforms the HMM-GMM approach (0.299). The highest CCA is 0.554, which is achieved by the combination of FBank and the two prosodic features (energy and F0). The FBank_E_F0 network has 120 units in the hidden layer. This suggests that head motion not only reflects paralinguistic information but also conveys important linguistic information as E and F0 are more related to the paralinguistic information and FBank encodes the linguistic information.

5.3. Performance of Acoustic Context

Previous neural network approaches for speech recognition have shown that long context acoustic input can benefit to the performance [22]. Moreover, a distinct head motion may last for 0.4-0.8s [10]. This motivates us to investigate the best acoustic context for head motion prediction. We test the context window with different frames from 11 to 31 with an interval of 4 frames. A generatively-pretrained 1-layer network is used and FBank_E_F0 is adopted as the network input. For

[1] For F0 and E, first and second derivatives are included.

Table 4: The performance of different acoustic context.

Context	11	15	19	23	27	31
CCA	0.554	0.549	**0.561**	0.558	0.552	0.547

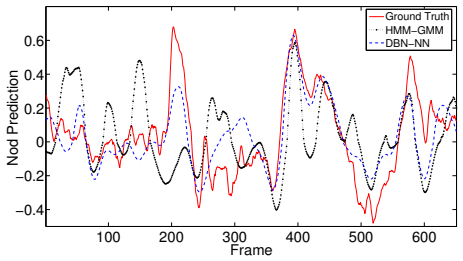

Figure 3: Predicted nod (Normalized) trajectories for DBN-NN and HMM-GMM

different context configuration, the number of hidden layer units is tuned for best performance. The results in Table 4 show that the acoustic feature span of 19 frames achieves the best performance and the best-performed network has 160 units in the hidden layer. Further increasing the context cannot bring performance gain.

5.4. Extra Pretraining Data

We examine whether the extra unlabeled data is beneficial to the head motion prediction in the network pre-training stage. Extra 73-minute speech data from the same anchor is added to the pre-training phase. The feature combination FBank_E_F0 is used and a context window of 19 frames creates an input layer of 532 visible units for the network. The network has one hidden layer and 160 units. Results show that with the help of the extra pre-training data, CCA is increased from 0.561 to 0.565. This performance gain may suggest that the generative pre-training phase achieves better network initialization with more acoustic data. Figure 3 shows the head nod trajectories generated by different approaches for a test clip. We can see that the trajectory generated by our DBN-NN approach is much closer to the ground truth trajectory.

6. Conclusions

In this paper, we address the speech-driven head motion synthesis problem by neural networks. Through a learned neural network from audio-visual data, our approach can predict head movement of a speaker from his/her speech. We have investigated the problem through four important aspects: the training strategy, the ideal structure of the network, the most effective acoustic features and the best context of acoustic feature input. Our study leads to several important conclusions. First, a generatively pre-trained neural network significantly outperforms a conventional randomly initialized network and the HMM-GMM approach [10, 11] in head motion prediction. Second, the feature combination of FBank, energy and F0 has the best ability in head motion synthesis. Third, a relatively longer acoustic feature context (19 frames) shows better head motion synthesis performance. In addition, extra training data in the pre-training stage can further improve the head motion prediction performance. In summary, the proposed approach increases the CCA from 0.299 (HMM-GMM) to 0.565 and it can be used to drive natural head motions for talking avatar applications. In the future, we plan to extend our work to head motion synthesis for multiple speakers. We also consider to test the synthesis performance subjectively through a talking avatar.

7. References

[1] K. G. Munhall, J. A. Jones, D. E. Callan, T. Kuratate, and E. Vatikiotis-Bateson, "Visual prosody and speech intelligibility head movement improves auditory speech perception," *Psychological Science*, vol. 15, no. 2, pp. 133–137, 2004.

[2] E. Cosatto, J. Ostermann, H. P. Graf, and J. Schroeter, "Lifelike talking faces for interactive services," *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1406–1429, 2003.

[3] C. Busso and S. S. Narayanan, "Interrelation between speech and facial gestures in emotional utterances: A single subject study," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 8, pp. 2331–2347, 2007.

[4] C. Dehon, P. Filzmoser, and C. Croux, "Robust methods for canonical correlation analysis," in *Data Analysis, Classification, and Related Methods*. Springer, 2000, pp. 321–326.

[5] C. Busso, Z. Deng, U. Neumann, and S. Narayanan, "Natural head motion synthesis driven by acoustic prosodic features," *Computer Animation and Virtual Worlds*, vol. 16, no. 3-4, pp. 283–290, 2005.

[6] S. Zhang, Z. Wu, H. M. Meng, and L. Cai, "Head movement synthesis based on semantic and prosodic features for a chinese expressive avatar," in *Proc. ICASSP*. IEEE, 2007, pp. 837–840.

[7] K. Mu, J. Tao, J. Che, and M. Yang, "Mood avatar: Automatic text-driven head motion synthesis," in *Proc.ICMI-MLMI*. ACM, 2010, p. 37.

[8] C. Busso, Z. Deng, M. Grimm, U. Neumann, and S. Narayanan, "Rigid head motion in expressive speech animation: Analysis and synthesis," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1075–1086, 2007.

[9] H. P. Graf, E. Cosatto, V. Strom, and F. J. Huang, "Visual prosody: Facial movements accompanying speech," in *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, 2002, pp. 396–401.

[10] G. Hofer and H. Shimodaira, "Automatic head motion prediction from speech data." in *Proc. INTERSPEECH*, 2007, pp. 722–725.

[11] G. Hofer, H. Shimodaira, and J. Yamagishi, "Speech driven head motion synthesis based on a trajectory model," in *Proc. SIGGRAPH*, 2007.

[12] A. Ben Youssef, H. Shimodaira, and D. A. Braude, "Articulatory features for speech-driven head motion synthesis," in *Proc. INTERSPEECH*, Lyon, France, August 2013, pp. 2758–2762.

[13] A. Ben Youssef, H. Shimodaira, and D. Braude, "Speech driven talking head from estimated articulatory features," in *Proc. ICASSP*, Florence, Italy, May 2014, pp. 4606–4610.

[14] M. E. Sargin, Y. Yemez, E. Erzin, and A. M. Tekalp, "Analysis of head gesture and prosody patterns for prosody-driven head-gesture animation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1330–1345, 2008.

[15] H. C. Yehia, T. Kuratate, and E. Vatikiotis-Bateson, "Linking facial animation, head motion and speech acoustics," *Journal of Phonetics*, vol. 30, no. 3, pp. 555–568, 2002.

[16] B. Li, L. Xie, and P. Zhu, "Head motion generation for speech driven talking avatar," *Journal of Tsinghua Univ (Sci & Tech)*, vol. 53, no. 6, pp. 898–902, 2013.

[17] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[18] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing," *Signal Processing Magazine, IEEE*, vol. 28, no. 1, pp. 145–154, 2011.

[19] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

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

[21] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[22] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams *et al.*, "Recent advances in deep learning for speech research at microsoft," in *Proc. ICASSP*. IEEE, 2013, pp. 8604–8608.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Representations by Back-Propagating Errors*. MIT Press, Cambridge, MA, USA, 1988.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AI & Statistics*, 2010, pp. 249–256.

[25] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[26] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[27] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, "The HTK book (for HTK version 3.4)," *Cambridge university engineering department*, vol. 2, no. 2, pp. 2–3, 2006.

[28] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, "Tandem-straight: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, f0, and aperiodicity estimation," in *Proc. ICASSP*. IEEE, 2008, pp. 3933–3936.

[29] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proc. CVPR*. IEEE, 2013, pp. 532–539.

[30] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. of Sixth ISCA Workshop on Speech Synthesis*, 2007, pp. 294–299.