Multimed Tools Appl (2015) 74:9889–9907
DOI 10.1007/s11042-014-2183-z

Acoustic to articulatory mapping with deep neural network

Zhiyong Wu · Kai Zhao · Xixin Wu · Xinyu Lan · Helen Meng

Received: 28 February 2014 / Revised: 4 June 2014 / Accepted: 7 July 2014 /
Published online: 1 August 2014

Abstract Synthetic talking avatar has been demonstrated to be very useful in human-computer interactions. In this paper, we discuss the problem of acoustic to articulatory mapping and explore different kinds of models to describe the mapping function. We try general linear model (GLM), Gaussian mixture model (GMM), artificial neural network (ANN) and deep neural network (DNN) for the problem. Taking the advantage of neural network that its prediction stage can be finished in a very short time (e.g. real-time), we develop a real-time speech driven talking avatar system based on DNN. The input of the system is acoustic speech and the output is articulatory movements (that are synchronized with the input speech) on a three-dimensional avatar. Several experiments are conducted to compare the performance of GLM, GMM, ANN and DNN on a well known acoustic-articulatory

Z. Wu · K. Zhao · X. Wu · X. Lan · H. Meng
Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems, and Shenzhen Key Laboratory of Information Science and Technology, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China

Z. Wu
e-mail: zywu@se.cuhk.edu.hk

K. Zhao
e-mail: zk69052@163.com

X. Lan
e-mail: 451250406@qq.com

H. Meng
e-mail: hmmeng@se.cuhk.edu.hk

Z. Wu · H. Meng
Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Hong Kong, SAR, China

Z. Wu · K. Zhao · X. Wu (✉) · X. Lan
Tsinghua National Laboratory for Information Science and Technology (TNList),
and Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
e-mail: xixinwood@gmail.com

English speech corpus MNGU0. Experimental results demonstrate that the proposed acoustic to articulatory mapping method with DNN can achieve the best performance.

Keywords Acoustic to articulatory mapping · Audio-visual mapping · Deep neural network (DNN) · Speech driven talking avatar

1 Introduction

Human speech is bimodal in nature. While audio is the major source of speech information, visual component is considered to be valuable supplementary in noisy environments because it remains unaffected by acoustic noise. Synthetic talking avatar, with human-like appearance and articulator movement synchronized with speech, has been demonstrated to be very useful in human-computer interaction applications [2, 4, 11, 10, 16, 26, 28] such as computer agent, virtual newscaster, email reader, information kiosk, etc. As has been discussed in many research works, a synthetic talking face has much to offer in addition to acoustic speech [1, 12], for example to help people understand related speech in noisy environment [14], to provide an aid for the hearing-impaired where the simulated lip movements can help the user decipher the acoustic speech.

The purpose of our work is to develop a real-time speech driven talking avatar system [32] based on deep neural network. The input of the system is the acoustic speech and the output is the real-time generated articulatory movement animation on a virtual talking avatar. The generated movements of the articulators (e.g. lips, tongue, velum, etc.) are synchronized with the input speech. In speech production, there are direct connections between the configurations of the articulators, which are the positions and movements of the lips, tongue, velum, etc., and the speech. In speech driven talking avatar, the most important issue is acoustic to articulatory mapping. In another words, we expect to develop a mapping function between the acoustic speech and the articulatory movement. The input of the mapping function is the features representing the acoustic speech and the output is the articulatory features.

However, it is not trivial to model such mapping procedure because the relationship between the acoustic and articulatory features is a non-linear and not a one-to-one mapping. Furthermore, the articulator movements are always determined by not only the current pronounced phoneme, but also the succeeding or preceding phonemes (so called the coarticulation phenomenon). To solve this challenging problem, lots of methods and models have been proposed and huge improvements have been achieved during the past decades.

To capture the mapping relation between acoustic and articulatory features, the simplest method is the linear mapping, for example the general linear model (GLM) [15]. But because the acoustic to articulatory mapping is actually non-linear, the linear model cannot achieve idea performance. Hidden Markov model (HMM) is then proposed to tackle the problem [8, 9, 30, 31]. In this method, the correspondence between the acoustic and articulatory features is described as a linear mapping in each state of the HMM; and the phonetic information is required for training the HMM and used as constraints to address the one-to-many mapping problem. Gaussian mixture model (GMM) is used to model the joint distribution of acoustic and articulatory features based on a parallel acoustic-articulatory speech corpus [23, 24]. This model can address the mapping without constraints on phonetic information as in HMM. A dynamic Bayesian network based audio-visual articulatory model was proposed in [27] to model the correlation between audio and video features, and Baum-Welch inversion algorithm was presented to generate optimal facial parameters from audio with the proposed model. Although the dynamic Bayesian network with Baum-Welch inversion algorithm can achieve realistic mouth-synching, the recursive steps for computing the optimal articulatory features has prevent the method from being used in the real-

time applications. With the development of backpropagation method [21], multilayered neural networks with hidden layers have attracted many research interests. The artificial neural network (ANN) is also adopted in solving the acoustic to articulatory mapping problem [19]. Although the training of ANN is time consuming, it takes little time to compute the output from the input for ANN, which is an excellent property in building the real-time system for speech driven talking avatar. Hence, ANN has demonstrated superior feature over other models. However, the training of ANN is not trivial as different initialization of the ANN weights may leads to different training results. Generally, the ANN should be trained with several initializations and the best ANN is selected among these different weight sets.

In this paper, we describe our work with the effective deep neural network (DNN) [6, 7] to model the acoustic to articulatory mapping for real-time speech driven talking avatar system. Why we choose DNN is because it has shown many superior characteristics over traditional ANN [3]. First, the unsupervised pre-training step of the DNN can make effective use of large amount of unlabeled training data. Second, the pre-training step provides a good initialization point for the neural network which can overcome the traditional ANN method where different initialization sets should be tried. Third, the over-fitting problem of the traditional ANN can be effectively addressed by the pre-training step. Furthermore, unlike [25] where conventional contrastive divergence (CD) [4] has been used for the pre-training of DNN, we use the persistent contrastive divergence (PCD) algorithm [22] which leads to performance improvement for acoustic to articulatory mapping. We will conduct several experiments to explore the performances of different models, including GLM, GMM, ANN and DNN. We also introduce our work of building a real-time speech driven talking avatar system based on DNN.

The rest of the paper is organized as follows. Section 2 introduces the related work in this area, including GLM, GMM and ANN. Section 3 describes our work on acoustic to articulatory mapping (audio-visual mapping) with DNN. We will illustrate the main principle of DNN, its advantages and the use of DNN for audio-visual mapping in real-time speech driven talking avatar. Experiments and results are then presented in Section 4. Finally, Section 5 lays out the conclusions.

2 Related work

To explore the audio-visual mapping correlation between the acoustic speech and the articulatory movement, lots of researches have been devoted in the past decades and several models have been proposed to model the relations, including general linear model (GLM), Gaussian mixture model (GMM) and artificial neural network (ANN). We will give a brief introduction on these models in this section.

2.1 General linear model (GLM)

General linear model (GLM) [15] is a simple way to model the mapping between the acoustic and articulatory features. It assumes that the relation between the acoustic speech and the articulatory movement is linear (though it is demonstrated by our experiments that it is not appropriate to make such an assumption). The GLM can be expressed as the following:

$$\mathbf{y} = k\mathbf{x} + b, \tag{1}$$

where $\mathbf{y}$ is the vector of target value in the mapped range while $\mathbf{x}$ is the input vector in the mapping domain, k and b are the parameters to be estimated. To find out the best parameters k and b for GLM, the most general way is to apply the least square method (LSM). Assume that

we have a series of input values $\mathbf{x}=\{x_1,x_2,\ldots,x_n\}$ and corresponding target values $\mathbf{y}=\{y_1,y_2,\ldots,y_n\}$, where n is the number of training samples. By denoting the summation of squares as S:

$$\mathrm{S} = [y_1-(kx_1+b)]^2 + [y_2-(kx_2+b)]^2 + \cdots + [y_n-(kx_n+b)]^2, \tag{2}$$

the purpose is to estimate the parameters k and b by minimizing the value of S. It is simple to find the optimal value of k and b by taking the partial derivatives of k and b respectively.

For the multi-dimensional input and target value, it is easy to extend k and b to multi-dimension and the above operation in equation (1) becomes a kind of matrix transformation. In acoustic to articulatory mapping, the input acoustic features are always multi-dimensional, e.g. in our work, the dimension of the input acoustic feature is 451 and that of the output articulatory feature is 36 (The meaning of each dimension will be elaborated in Section 4).

2.2 Gaussian mixture model (GMM)

Gaussian mixture model (GMM) [18] is a statistical probabilistic model with probability density function represented as a weighted sum of several Gaussian component densities. Since GMM can smoothly approximate density distribution in any shapes, it is widely used in different research areas of speech processing, such as speech recognition, speaker recognition, speech synthesis, etc.

While applying GMM for the acoustic to articulatory mapping problem, the mapping function can be defined as:

$$\widehat{\mathbf{y}}_t = \sum_{i=1}^{M} p(m_i|\mathbf{x}_t,\Theta)\mathbf{E}(\mathbf{y}_t|\mathbf{x}_t,m_i,\Theta), \tag{3}$$

Where

$$\mathbf{E}(\mathbf{y}_t|\mathbf{x}_t,m,\Theta) = \boldsymbol{\mu}_i^{(y)} + \sum\nolimits_i^{(yx)} \sum\nolimits_i^{(xx)^{-1}} \left(\mathbf{x}_t - \boldsymbol{\mu}_i^{(x)}\right), \tag{4}$$

$$p(m_i|\mathbf{x}_t,\Theta) = \frac{w_i N\left(\mathbf{x}_t;\boldsymbol{\mu}_i^{(x)},\sum\nolimits_i^{(xx)}\right)}{\sum_{j=1}^{M} w_j N\left(\mathbf{x}_t;\boldsymbol{\mu}_j^{(x)},\sum\nolimits_j^{(xx)}\right)}. \tag{5}$$

In the above equations, we tempt to model the mapping function from the acoustic feature vector $\mathbf{x}_t$ to the articulatory feature vector $\mathbf{y}_t$ in frame t. $\widehat{\mathbf{y}}_t$ is the estimated articulatory feature vector, M is the total number of Gaussian mixtures (i.e. Gaussian components). Θ is the set of parameters of the model including weights of Gaussian components, mean vectors of Gaussian components and covariance matrices in the joint Gaussian distribution. Assume that w_i is the weight of the i-th Gaussian mixture, $\boldsymbol{\mu}_i^{(x)}$ and $\boldsymbol{\mu}_i^{(y)}$ are the mean vector of the i-th mixture of $\mathbf{x}$ and $\mathbf{y}$ respectively, and denote the covariance matrix of the i-th mixture for $\mathbf{x}$ and the cross-covariance matrix for the i-th mixture for $\mathbf{x}$ and $\mathbf{y}$ as $\boldsymbol{\Sigma}_i^{(xx)}$ and $\boldsymbol{\Sigma}_i^{(xy)}$. $N(\mathbf{x}_t;\boldsymbol{\mu}_i^{(x)},\boldsymbol{\Sigma}_i^{(xx)})$ is the normal distribution with mean vector $\boldsymbol{\mu}_i^{(x)}$ and covariance matrix $\boldsymbol{\Sigma}_i^{(xx)}$. As shown in the equations, the estimated articulatory feature vector $\widehat{\mathbf{y}}_t$ is simply a linear mixture of several Gaussian distributions. The parameters of the Gaussian mixtures can be figure out by training.

2.3 Artificial neural network (ANN)

Artificial neural network (ANN) [29] is an interconnected group of nodes that are called artificial neurons. Between the nodes, there are arrows representing connections from the output of one neuron to the input of another. The adaptive weights are defined as the connection strengths between neurons, and are activated during training and prediction process.

A simple three layers artificial neural network includes input layer, hidden layer and output layer. The input layer consists of input neurons which send data from the input to the hidden layer neurons, and then the hidden layer neurons send data to the output layer neurons. More complex networks will have more hidden layers of neurons with similar working principle. The parameters called "weights" between two neurons play a key role in manipulating the data in the calculations.

Typically, an ANN is defined by these parameters: the network structure, the learning process for updating the weights and the activation function that converts a neuron's weighted input to its output. Mathematically, a neuron's activation function is defined as:

$$o = \sum_{i=0}^{n} w_i x_i. \tag{6}$$

When it comes to the classification problem, we want the output to be discrete values, and the activation function can be realized with the following function:

$$o = \begin{cases} 1, & if \sum_{i=0}^{n} w_i x_i > 0 \\ -1, & otherwise \end{cases}. \tag{7}$$

The goal of training the ANN is to derive a set of weights w_i (i=0,…,n) that can minimize the mean square error (MSE) $E(w)$ which measures the differences (i.e. errors) between the output o_d of ANN and the target value t_d. Hence, the MSE can be calculated as:

$$E(w) = \frac{1}{2} \sum_{d=1}^{D} (t_d - o_d)^2, \tag{8}$$

where D is the dimension of the output. We minimize this error using gradient descent for the class of neural networks, and update the weights. Let x_i be the input value of the input node i, and η be the learning rate, then the updated weights can be calculated as:

$$w_i \leftarrow w_i + \Delta w_i, \tag{9}$$

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} = \eta (t_i - o_i)(1 - o_i) x_i. \tag{10}$$

3 Audio-visual mapping with deep neural network

Although ANN can achieve great performance improvement in solving the acoustic to articulatory mapping problem, we found in our experiment, when the number of the layer of

ANN increases, the neural network tends to get over-fitted. Deep neural network (DNN) has shown many superior characteristics over traditional ANN. The over-fitting problem of the traditional ANN can be effectively addressed by the unsupervised pre-training step in DNN by making sufficient use of large amount of unlabeled training data.

3.1 Deep belief network (DBN)

It is well known that it is hard to train a deep neural network (DNN) directly, for the highly non-convex property, gradient diffusion and pathological curvature of the training problem. The deep belief network (DBN) was the first solution proposed to this difficult problem [7, 17]. With DBN, a DNN can be trained effectively and huge number of data can be used to train the network leading to performance improvement.

3.1.1 Restricted boltzmann machine (RBM)

A DBN can be trained as a stack of restricted Boltzmann machines (RBMs) [5], in which each two neighbor layers is considered as an RBM. An RBM is a probabilistic model represented by an undirected graphical model, in which there are two layers of probabilistic units, i.e. a hidden/latent variable layer and a visible variable layer. As Boltzmann machine (BM), every unit in hidden layer is fully connected to the units in visible layer and vice verse. But unlike BM, in RBM, the units in the same layer are not connected to each other. Denoting the hidden variable layer and visible variable layer as **v** and **h** respectively, we assign each hidden and visible layer pair (**v**, **h**) with an energy function $E(\mathbf{v},\mathbf{h})$. The joint probability distribution of **v** and **h** can be modeled as:

$$p(\mathbf{v},\mathbf{h}) = \frac{1}{Z}e^{-E(\mathbf{v},\mathbf{h})}, \tag{11}$$

where $Z=\sum_{\mathbf{vh}}e^{-E(\mathbf{v},\mathbf{h})}$ is a normalization factor.

For different value type of **v** and **h**, the RBM has different attribute. If both **v** and **h** are multidimensional binary variables, a Bernoulli-Bernoulli RBM will be used. In this case, the energy function can be typically defined as:

$$E(\mathbf{v},\mathbf{h}) = -\mathbf{a}^T\mathbf{v}-\mathbf{b}^T\mathbf{h}-\mathbf{v}^T\mathbf{W}\mathbf{h}, \tag{12}$$

where **W** is a matrix of weights between **v** and **h**, **a** and **b** is the bias vector of visible layer and hidden layer respectively.

For the problems with real-valued input variable **v**, and **h** is still multidimensional binary variable, a Gaussian-Bernoulli RBM can be used. The energy function can be typically defined as:

$$E(\mathbf{v},\mathbf{h}) = -\frac{1}{2}(\mathbf{a}-\mathbf{v})^T(\mathbf{a}-\mathbf{v})-\mathbf{b}^T\mathbf{h}-\mathbf{v}^T\mathbf{W}\mathbf{h}. \tag{13}$$

Conventionally, the input variables to Gaussian-Bernoulli RBM are usually normalized over the training data to have mean 0 and standard deviation 1.

3.1.2 Stacked RBM

DBN is a multi-layer generative probabilistic model. Actually, it is quite simple to acquire a DBN using a series of trained RBMs by stacking the RBMs together one by one in series. In

the stacked RBMs, the visible layer of the former RBM serves as the hidden layer of the later RBM. In this way, we can get a multi-layer generative probabilistic model with one visible layer and many hidden layers. The model is called deep belief network (DBN). Though the method is simple, it is powerful and such greedy training fashion [17] has been proved that the variational lower bound of the probability of visible variable can be guaranteed.

3.2 Deep neural network (DNN)

Deep neural network (DNN) is actually a feed forward neural network (also called multi-layer perceptron, MLP) with many hidden layers. The number of the layers is conventionally between 2 and 10. Assume that we have a DNN with K layers (excluding the input visible layer), and denoting the weight matrix and hidden bias from bottom to up as $\mathbf{W}_k$ and $\mathbf{b}_k$, where k=1,2,…,K, and $h_{ki}(\mathbf{x})$ is the output of the i-th neuron in hidden layer k, $\mathbf{h}_k(\mathbf{x})=[h_{ki}(\mathbf{x}),\ i=1,2,\ldots,I]^T$ is the output vector of hidden layer k, where I is the number of neurons in hidden layer k, then we have

$$\mathbf{h}_k(\mathbf{x}) = sigmoid(\mathbf{u}_k(\mathbf{x})), k = 1, 2, \ldots, K, \tag{14}$$

where $\mathbf{x}$ is the input value vector, and

$$\mathbf{u}_k(\mathbf{x}) = \mathbf{W}_k\mathbf{h}_{k-1}(\mathbf{x}) + \mathbf{b}_k, \tag{15}$$

$$sigmoid(x) = \frac{1}{1+e^{-x}}. \tag{16}$$

Here, $\mathbf{h}_0(\mathbf{x})$ is the input variable $\mathbf{h}_0(\mathbf{x})=\mathbf{x}$, and $\mathbf{u}_K(\mathbf{x})$ is the desired output $\mathbf{u}_K(\mathbf{x})=\mathbf{y}$. With the DNN, suppose we can get an output $\widetilde{\mathbf{y}}$ from the input $\mathbf{x}$, to train the DNN, the output value $\widetilde{\mathbf{y}}$ of the DNN is expected to approach the given target data $\mathbf{y}$ as much as possible. Hence, the learning of DNN is done by optimizing the following loss function as:

$$L\left(\mathbf{y},\widetilde{\mathbf{y}}\right) = \left\|\mathbf{y}-\widetilde{\mathbf{y}}\right\|_2^2. \tag{17}$$

3.2.1 Pre-training

The purpose of pre-training is to derive a set of weights that can be served as the initialization of the weights for DNN for later fine-tuning. The similar structure between DBN and feed forward DNN has made it quite natural to utilize the weights learnt in a DBN to provide new initializing weights for DNN other than the random small Gaussian weights traditionally used to train a neural network. What's more, with such a pre-training procedure, the network can converge faster and have better convergence results, and this makes the training of DNN much easier.

Unlike the traditional method of using contrastive divergence (CD) [4] to train RBMs for DBN, we use the persistent contrastive divergence (PCD) algorithm [22]. The idea of PCD algorithm is that when sampling from the RBM, instead of re-initializing the running of Markov chain during each epoch, the PCD algorithm initializes the Markov chain by utilizing the state obtained from last epoch, and moves it one step forward as an approximation of the sample from the model. Thus, the sample would be closer to the real model distribution after each epoch and yet the amount of calculation needed is almost as same as CD.

3.2.2 Fine-tuning with back-propagation

After pre-training, a DNN can be trained with traditional back-propagation algorithm just as an MLP. The fine-tuning back-propagation procedure can be defined as follows:

$$\frac{\partial L\left(\mathbf{y},\widetilde{\mathbf{y}}\right)}{\partial \mathbf{u}_K(\mathbf{x})} = -2\left(\mathbf{y}-\widetilde{\mathbf{y}}\right). \tag{18}$$

and for every $k=K$-1,…,2,1,

$$\frac{\partial L\left(\mathbf{y},\widetilde{\mathbf{y}}\right)}{\partial u_{ki}(\mathbf{x})} = \frac{\partial L}{\partial h_{ki}(\mathbf{x})} h_{ki}(\mathbf{x})(1-h_{ki}(\mathbf{x})), \tag{19}$$

$$\frac{\partial L\left(\mathbf{y},\widetilde{\mathbf{y}}\right)}{\partial \mathbf{u}_K(\mathbf{x})} = \mathbf{W}_{k+1}^T \frac{\partial L}{\partial \mathbf{u}_{k+1}(\mathbf{x})}. \tag{20}$$

3.3 DNN for audio-visual mapping

In the task of acoustic to articulatory mapping (audio-visual mapping), the input of the DNN is the real-valued acoustic features and the output is the values of the articulatory features (i.e. articulator positions). In our work, a Gaussian-Bernoulli RBM is used for the bottom two layers of the DNN, and each dimension of the acoustic feature input is normalized over the training set to have mean 0 and standard variance 1. The most top layer of DNN is a linear regression layer, in which each output unit corresponds to one articulator position to infer.

Taking the advantage of DNN for audio-visual mapping, we develop a real-time speech driven talking avatar system based on DNN. The architecture of the proposed system is illustrated in Fig. 1, where the audio-visual mapping function is achieved by virtue of DNN.

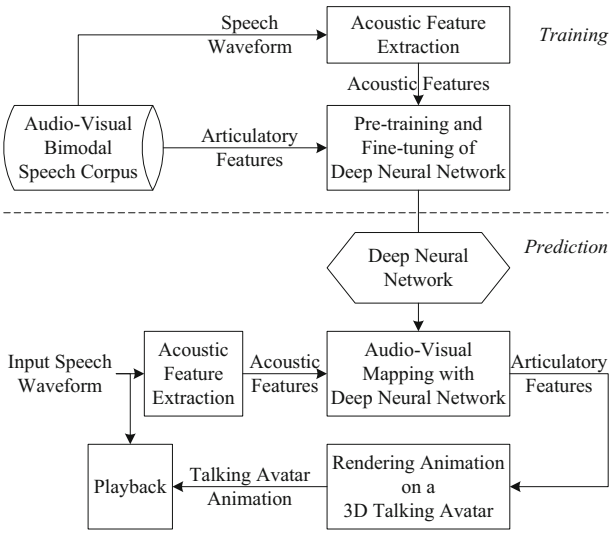

Fig. 1 The architecture of the proposed real-time speech driven talking avatar system, where acoustic to articulatory mapping is achieved by incorporating deep neural network

Two stages are involved in the proposed speech driven talking avatar system: the training stage and the prediction stage. During training stage, speech waveforms from the training audio-visual bimodal speech corpus are fed to the acoustic feature extraction module to extract the acoustic features for training the DNN. Articulatory features are also extracted from the training bimodal corpus. Pre-training technology is then utilized to provide a good initialization point for the parameters of the DNN by training it as stacked RBMs. After training the stacked RBMs, all the units and weights are treated like a traditional neural network to perform fine-tuning to get better regression performance.

During prediction stage, the acoustic features of the input speech waveform are also extracted by the acoustic feature extraction module. These acoustic features are served as the input of the DNN for audio-visual mapping. The articulatory features output from the DNN are sent to a 3D talking avatar rendering module to generate the talking avatar animation, which is finally playback together with the input speech.

4 Experiments

To evaluate the performance of the proposed approach, we conduct a set of experiments on acoustic to articulatory mapping with different models including GLM, GMM, ANN and DNN. The experimental results validate that DNN can achieve the best performance in solving the task.

4.1 Database

We have used the electromagnetic articulography (EMA) dataset of the MNGU0 corpus [20] as the database for all the acoustic to articulatory mapping experiments with different models. The MNGU0 corpus uses the Cartsens AG500 electromagnetic articulograph, in which 6 transmitter coils are used to track the positions of the 6 articulators in the midsagittal plane: 3 on the tongue, one on the lower incisor and one each on the upper and lower lips, including upper lip (UL), lower lip (LL), lower incisor (LI), tongue tip (T1), tongue blade (T2) and tongue dorsum (T3). Fig. 2 shows the positions of 6 coils used in tracking the articulator positions.

MNGU0 corpus consists of two parts. The first part is the articulator position data (i.e. the visual articulatory data). When collecting the data, the x- and y-coordinates of the 6 coils in the

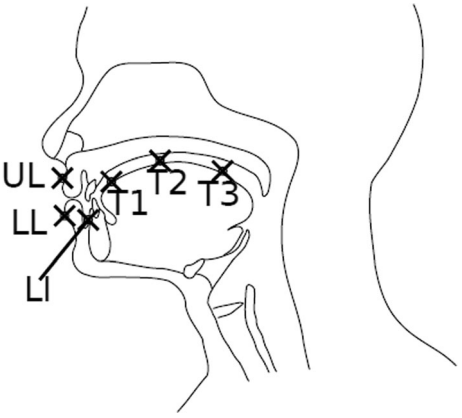

Fig. 2 Sample of the positions of the articulators tracked in the MNGU0 corpus (from [20]). Six EMA sensor coils are used to track the position of the articulators including upper lip (UL), lower lip (LL), lower incisor (LI), tongue tip (T1), tongue blade (T2), and tongue dorsum (T3)

midsagittal plane are recorded and used. So, the articulatory data used for our experiments include 12 channels of EMA data at sampling frequency of 200Hz.

The second part of MNGU0 corpus is the acoustic data. The original audio data has been converted to frequency warped line spectral frequencies (LSFs) of order 40 plus a gain value. The LSFs are derived from the spectral envelope estimated with STRAIGHT [13], with 5 msec frame shift to match the sampling rate of the EMA data. The initial and final silences have been removed.

Both EMA and LSF feature vectors were z-score normalized by subtracting their respective global mean and dividing by 4 times the standard deviation for each dimension.

The database contains 1,354 utterances and is divided into three subsets: a validation and test set each with 63 utterances, and a training set containing the rest 1,228 utterances [20].

4.2 Experimental setup

4.2.1 Performance measurement

To measure the performance of acoustic to articulatory mapping, root mean-squared error (RMSE) has been used as the performance measurement. It is defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}(e_i - t_i)^2}, \tag{21}$$

where e_i is the estimated (predicted) articulatory trajectory value and t_i is the actual measured articulatory value.

We calculate RMSE in each dimension of the articulatory data respectively, and then sum up the RMSEs of all 12 dimensions to get the final value of RMSE.

4.2.2 Experimental conditions

In our experiments, besides the 12 channels of EMA data, the first order and the second order differences (i.e. delta and acceleration) of the EMA data are also considered. Hence, the dimension of the final articulatory feature is $12 \times 3 = 36$. As for the acoustic data, a context window of 11 continuous acoustic frames (5 left frames, 1 current frame and 5 right frames) is used. Hence, the dimension of the final acoustic feature is $41 \times 11 = 451$.

For GMM, different number of mixtures including 4, 8, 16, 32 and 64 has been evaluated. Experiments indicate that the GMM with 32 Gaussian mixtures achieves the best performance. The experimental results reported in Section 4.3 are based on 32 mixtures.

Table 1 Configuration of the ANN

Parameter	Value
Learning rate	0.2
Max epochs	1,000
Momentum	0.9
Number of units for input layer	451
Number of units for output layer	36
Number of units for each hidden layer	100

Table 2 Configuration for pre-training of the input Gaussian-Bernoulli RBM layer of the DNN

Parameter	Value
Learning rate	0.001
Max epochs	10
Batch size	128
Momentum	0.9
Weight decay	0.001
Initial weights	$N(0,0.01)$
Number of units for visible layer	451

For ANN, 3 hidden layers are used, with each hidden layer contains 100 units (i.e. neurons). For the input layer, it contains 451 units related to the acoustic feature. For the output layer, it contains 36 units corresponding to the articulatory data. During the training of ANN, the learning rate has been set to be multiplied by 0.2, and the momentum is set to be 0.9. The configuration of all the parameters of the network is listed in Table 1.

For DNN, also 3 hidden layers are used. The input layer is Gaussian-Bernoulli RBM, and all the other layers are Bernoulli-Bernoulli RBMs. During pre-training, the configurations of the input RBM layer and other hidden RBM layers are shown in Tables 2 and 3 respectively. Please be noted that, for both types of layers, the learning rate will be multiplied by 0.998 after each iteration; the momentum in the first 20 epochs increases evenly and remains at 0.9 after the 20th epoch. As for fine-tuning, the configuration is shown in Table 4. Same as the above, the learning rate will be multiplied by 0.998 after each iteration. However, the momentum increases evenly in the first 10 epochs and remains at 0.99 thereafter. With the same configuration as stated here, we conduct 5 experiments with 100, 200, 300, 400 and 500 units for each hidden layer in the DNN architecture, and the results are shown in the following section.

4.3 Experimental results

4.3.1 Experiment on different models

Several acoustic to articulatory mapping experiments have been conducted for different models including GLM, GMM, ANN and DNN. For each experiment, the RMSE is calculated and used as the performance measurement to compare different models. The results are shown in Table 5.

Table 3 Configuration for pre-training of the Bernoulli-Bernoulli RBM layers of the DNN

Parameter	Value
Learning rate	0.01
Max epochs	5
Batch size	128
Momentum	0.9
Weight decay	0.0001
Initial weights	$N(0,0.01)$
Number of units for visible layer	(100,200,300,400,500)

Table 4 Configuration for fine-tuning of the DNN

Parameter	Value
Learning rate	0.001
Max epochs	5,000
Batch size	128
Momentum	0.99
Weight decay	0.0002

As can be seen, the RMSE is 1.92 mm for GLM, and 2.14 mm for GMM. While for ANN with 100 units per hidden layer, the RMSE is 1.04 ms. And the RMSE is 0.67 ms for the model of DNN with 100 units per hidden layer. The results demonstrate that the use of DNN for acoustic to articulatory mapping can achieve the best performance. It should also be noted that, different from expectation, the GMM model performs the worst according to the RMSE measurement. This might be because in our experiments, we have used diagonal matrix instead of full matrix for the cross-covariance matrices in GMM.

4.3.2 Experiment on DNN with different parameter configurations

We further conduct experiments to evaluate the performance of DNN with different parameter configurations.

The results are shown in Fig. 3, where the x-axis indicates the number of fine-tuning iteration epochs for the DNN training while the y-axis shows the generation error (RMSE) of articulatory features. As can be seen, the RMSE declines along with the iteration epochs during the entire training procedure of DNN for different number of hidden units (i.e. number of the units for each hidden layer). It should be noted that, for the configuration of 100, 200, 300, 400 and 500 hidden units, the RMSE error declines when the number of hidden units increases from 100 to 400. However, the RMSE error increases dramatically when the number of hidden units increases to 500. This result indicates that increasing the number of hidden units is helpful for performance improvement of the DNN in the task of acoustic to articulatory mapping. However, when the number of hidden units exceeds some threshold, the performance of DNN may degrade a lot.

It can also be seen that the RMSE value remains steady when the number of epochs is greater than a threshold, and such threshold might vary for different number of hidden units. The RMSE values at the threshold epoch on the test set with different number of units for hidden layers are shown in Table 6, where the related threshold numbers of epochs are also listed. As can be seen from Table 6 and Fig. 3, although the final performance of DNN with 200, 300 and 400 hidden units are similar, the network converges the most quickly and achieves the best performance for the configuration of 400 units for each hidden layer.

Table 5 RMSE error on the test set of different models for acoustic to articulatory mapping

Model	RMSE (mm)
GLM	1.92
GMM	2.14
ANN (with 100 units for each hidden layer)	1.04
DNN (with 100 units for each hidden layer)	0.67

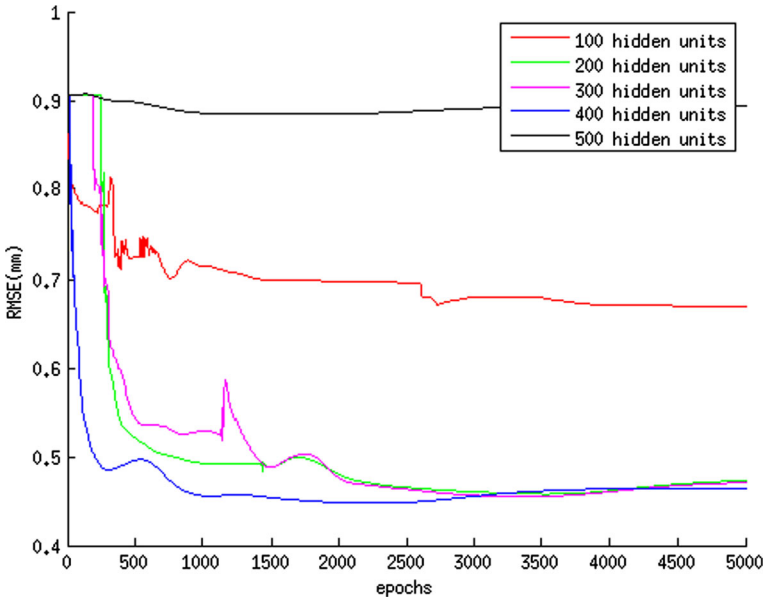

Fig. 3 RMSE error of the estimated articulatory features as a function of epochs during DNN training on the validation set

4.3.3 Experiment on articulatory trajectory prediction

Further experiments are conducted for the prediction of articulatory trajectory curves for different models of GLM, GMM, ANN and DNN. The results are shown in Fig. 4, which depicts the actually measured (red and bold curve, TEST) and the estimated articulatory trajectories for 200 frames of the tongue blade feature (T2) for all the four models to be compared.

As can be seen from the figure, the estimated articulatory trajectory generated from DNN (black and bold curve) is the closest one to the measured curve (TEST), which indicates that DNN achieves the best performance. It is in expectation that the GLM performs worse than ANN and DNN, as it uses the simplest linear mapping function and cannot capture the non-linear properties of the acoustic to articulatory mapping task very well. The performance of GMM is not so good, which is even worse than that of GLM. This is probably because in our experiments, we have used diagonal matrix instead of full matrix for the covariance matrices of the Gaussian mixtures in GMM. This can also explain why the estimated articulatory trajectory of GMM consists of lots of "horizontal bars" in Fig. 4 for the trajectory prediction experiment.

Table 6 RMSE error of DNN on the test set with different number of units for hidden layers

Number of units for hidden layers	Threshold epochs	RMSE(mm)
100	5,000	0.669
200	3,539	0.458
300	3,364	0.455
400	2,256	0.447
500	1,914	0.885

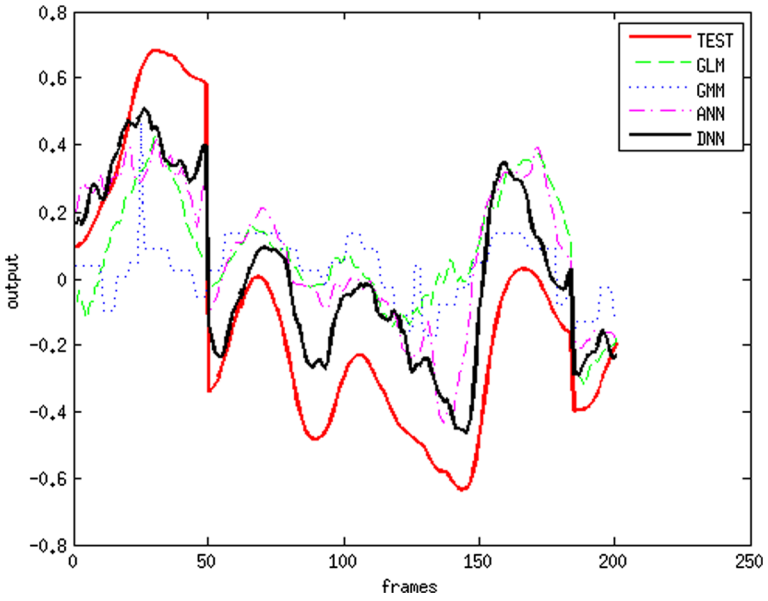

Fig. 4 Comparison between the actually measured (red and bold curve, TEST) and the estimated articulatory trajectories for 200 frames of the tongue blade feature (T2) for GLM, GMM, ANN and DNN, where DNN (black and bold curve) achieves the best performance

We can also find from the figure that the estimated values have shown some dynamic variances, while the actual measured curve is much smoother. This may be due to that DNN only performs regression from a context window of acoustic features to one frame of articulatory positions, and the continuity properties of the articulatory trajectories are not considered.

4.3.4 Experiment on the computational costs of different models

To compare real-time performance of different models, we further conduct experiments to measure the computational costs of predicting articulatory trajectories by different models including GLM, GMM, ANN and DNN. 100,000 frames (with 5 msec frame shift as described in section 4.1) of LSF features from the MNGU0 corpus are used as the input to the models. These frames amount to 100,000*5/1,000=500 s of acoustic input. The computational time of each model used to generate the articulatory trajectories from these acoustic parameters is

Table 7 Computational time of different models for predicting articulatory trajectories from 500 s of acoustic parameters (LSFs)

Model	Computational Time (s)
GLM	2
GMM	200
ANN (with 100 units for each hidden layer)	7
DNN (with 400 units for each hidden layer)	12

illustrated in Table 7. As can be seen, due to the complexity in computing the matrices, GMM takes the most time. For a second of input acoustic parameters, GMM will need 200/500=0.4 s to compute the trajectories. When compared to ANN, the computational time of DNN is about 1.72 times more than that of ANN because of the reason that more units (i.e. 400 units) per hidden layer are used in DNN while only 100 units per hidden layer for ANN. However, DNN can still perform acoustic to articulatory mapping in real-time; for a second of acoustic parameter input, DNN needs only 12/500=0.024 s to get the articulatory trajectories. GLM achieves the best computational performance because only linear transformation is needed.

5 Conclusions

In this paper, we discuss the problem of acoustic to articulatory mapping and perform several experiments to explore the mapping function. Since the mapping between the acoustic features and the articulatory movements is non-linear, we try four different kinds of models including GLM, GMM, ANN and DNN to validate their ability in describing the mapping relationship. It is in expectation that GLM performs not so well among the four types of models, since the problem is actually non-linear. However, GMM performs even worse than GLM. This might be due to the reason that, in principle assumption, the cross-covariance matrices of Gaussian components in GMM is full, while in our experiment it is set to diagonal. The performance of DNN is the best. Future work can be devoted to get smoother estimated articulator trajectories by considering the continuity properties of the articulator trajectories. We will also try to perform experiments on the GMM model with full covariance matrices and try to optimize the computation cost.

Acknowledgements This work is supported by the National Basic Research Program of China (2012CB316401 and 2013CB329304). This work is also partially supported by the Hong Kong SAR Government's Research Grants Council (N-CUHK414/09), the National Natural Science Foundation of China (61375027, 61370023 and 60805008), the National Social Science Foundation Major Project (13&ZD189) and Guangdong Provincial Science and Technology Program (2012A011100008).

References

1. Cassell J (2001) Embodied conversational agents: representation and intelligence in user interfaces. AI Mag 22(4):67–83
2. Cosatto E, Ostermann J, Graf HP, Schroeter J (2003) "Lifelike talking faces for interactive services". Proc IEEE 91:1406–1429
3. Deng L (2011) "An overview of deep-structured learning for information processing," In: Proc. Asian-Pacific Signal & Inforamtion Processing Annual Summit & Conference (APSIPA ASC), pp 1–14
4. Ding C, Xie L, Zhu PC (2014) Head motion synthesis from speech using deep neural networks. Multimed Tools Appl. doi:10.1007/s11042-014-2156-2
5. Hinton GE (2002) Training products of experts by minimizing contrastive divergence. Neural Comput 14(8): 1771–1800
6. Hinton GE (2007) To recognize shapes, first learn to generate images. Prog Brain Res 165:535–547
7. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7): 1527–1554

8. Hiroya S, Honda M (2002) "Determination of articulatory movements from speech acoustics using an HMM-based speech production model," In: Proc. Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pp 437–440
9. Hiroya S, Honda M (2002) "Acoustic-to-articulatory inverse mapping using an HMM-based speech production model," In: Proc. Int. Conf. on Spoken Language Processing (ICSLP), pp 2305–2308
10. Jia J, Wu ZY, Zhang S, Meng H, Cai LH (2013) Head and facial gestures synthesis using PAD model for an expressive talking avatar. Multimed Tools Appl. doi:10.1007/s11042-013-1604-8
11. Jia J, Zhang S, Meng FB, Wang YX, Cai LH (2011) Emotional audio-visual speech synthesis based on PAD. IEEE Transaction on Audio, Speech, and Language Processing, 19(3):570–582
12. Karlsson I, Faulkner A, Salvi G (2003) "SYNFACE - A talking face telephone," In: Proc. European Conf. on Speech Communication and Technology (EUROSPEECH), pp 1297–1300
13. Kawahara H, Estill J, Fujimura O (2001) "Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system straight," In: Proc. Int. Workshop Models and Analysis of Vocal Emissions for Biomedical Application (MAVEBA)
14. Massaro DW (1987) Speech perception by ear and eye: a paradigm for psychological inquiry. Lawrence Erlbaum Associates, Hillsdale
15. McCullagh P (1984) Generalized linear models. Eur J Oper Res 16(3):285–292
16. Meng FB, Wu ZY, Jia J, Meng H, Cai LH (2013) Synthesizing English emphatic speech for multimodal corrective feedback in computer-aided pronunciation training. Multimed Tools Appl. doi:10.1007/s11042-013-1601-y
17. Mohamed A, Dahl G, Hinton GE (2009) "Deep belief networks for phone recognition," In: Proc. NIPS Workshop on Deep Learning for Speech Recognition and Related Applications
18. Reynolds D (2009) "Gaussian mixture models," *Encyclopedia of Biometrics*
19. Richmond K (2002) "Estimating articulatory parameters from the acoustic speech signal," *PhD thesis*, The Centre for Speech Technology Research, Edinburgh University
20. Richmond K, Hoole P, King S (2011) "Announcing the electromagnetic articulography (day 1) subset of the MNGU0 articulatory corpus," In: Proc. Annual Conf. of International Speech Communication Association (INTERSPEECH), pp 1505–1508
21. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. Parallel Distrib Process 1:318–362
22. Tieleman T, Hinton GE (2009) "Using fast weights to improve persistent contrastive divergence," In: Proc. ACM International Conference on Machine Learning (ICML), pp 1033–1040
23. Toda T, Black AW, Tokuda K (2004) "Acoustic-to-articulatory inversion mapping with Gaussian mixture model," In: Proc. Annual Conf. of International Speech Communication Association (INTERSPEECH), pp 1129–1132
24. Toda T, Black AW, Tokuda K (2008) Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model. Speech Comm 50:215–227
25. Uria B, Murray I, Renals S, Richmond K (2012) "Deep architectures for articulatory inversion," In: Proc. Annual Conf. of International Speech Communication Association (INTERSPEECH)
26. Wu ZY, Zhang S, Cai LH, Meng H (2006) "Real-time synthesis of Chinese visual speech and facial expressions using MPEG-4 FAP features in a three-dimensional avatar," In: Proc. Int. Conf. on Spoken Language Processing (ICSLP), pp 1802–1805
27. Xie L, Liu ZQ (2007) Realistic mouth-synching for speech-driven talking face using articulatory modelling. IEEE Trans Multimedia 9(3):500–510
28. Xie L, Sun NC, Fan B (2013) A statistical parametric approach to video-realistic text-driven talking avatar. Multimed Tools Appl. doi:10.1007/s11042-013-1633-3
29. Yegnanarayana B (2006) *Artificial neural networks*, Prentice Hall of India
30. Zhang L, Renals S (2008) Acoustic-articulatory modeling with the trajectory HMM. IEEE Signal Process Lett 15:245–248
31. Zhao TY, Ling ZH, Lei M, Dai LR, Liu QF (2010) "Minimum generation error training for HMM-based prediction of articulatory movement," In: Proc. Int. Symposium on Chinese Spoken Language Processing (ISCSLP), pp 99–102
32. Zhao K, Wu ZY, Cai LH (2013)"A real-time speech driven talking avatar based on deep neural network," In: Proc. Asian-Pacific Signal & Inforamtion Processing Annual Summit & Conference (APSIPA ASC)

Zhiyong Wu Received the B.S. and Ph.D. degrees in computer science and technology from Tsinghua University, Beijing, China, in 1999 and 2005, respectively. He has been Postdoctoral Fellow in the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong (CUHK) from 2005 to 2007. He joined the Graduate School at Shenzhen, Tsinghua University, Shenzhen, China, in 2007, where he is currently an Associate Professor. He is also with the Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems. His research interests are in the areas of multimodal multimedia processing and communication, more specially, audiovisual bimodal modeling, text-to-audio-visual-speech synthesis, and natural language understanding and generation. Dr. Wu is a member of the Technical Committee of Intelligent Systems Application under the IEEE Computational Intelligence Society and the International Speech Communication Association.

Kai Zhao received the B.S. degree in physical science and engineering from Zhengzhou University, Henan, China, in 2011. He is now a master student in Tsinghua University. His main research interests include acoustic to articulatory inversion mapping and visual speech synthesis.

Xixin Wu Received the B.S. degree in software school from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2012. He is now a master student in Tsinghua University. His main research interests include natural language processing, sentiment analysis and expressive text-to-speech synthesis.

Xinyu Lan Received the B.S. degree in information and computing science from Jilin University, Jilin, China, in 2013. He is now a master student in Tsinghua University. He has been awarded national level prize of the National Undergraduate Innovative Experiment Program. His main research interests include expressive text-to-speech synthesis and artificial neural network technologies.

Helen Meng Received the S.B., S.M. and Ph.D. degrees, all in electrical engineering, from the Massachusetts Institute of Technology (MIT), Cambridge. She has been Research Scientist with the MIT Spoken Language Systems Group, where she worked on multilingual conversational systems. She joined The Chinese University of Hong Kong (CUHK) in 1998, where she is currently a Professor and Chairman in the Department of Systems Engineering and Engineering Management. In 1999, she established the Human-Computer Communications Laboratory at CUHK and serves as Director. In 2005, she established the Microsoft-CUHK Joint Laboratory for Human-Centric Computing and Interface Technologies, which was upgraded to MoE Key Laboratory in 2008, and serves as Co-Director. She is also Co-Director of the Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems. Her research interest is in the area of human–computer interaction via multimodal and multilingual spoken language systems, as well as translingual speech retrieval technologies. Prof. Meng has been elected IEEE Fellow in 2013 and Editor-in-Chief of the IEEE Transactions on Audio, Speech and Language Processing. She is also an elected board member of the International Speech Communication Association.