Hybrid Deep Neural Network - Hidden Markov Model (DNN-HMM) Based Speech Emotion Recognition

Longfei Li, Yong Zhao, Dongmei Jiang and Yanning Zhang
VUB-NPU Joint AVSP Research Lab
Northwestern Polytechnical University, Xi'an, China
Shaanxi Provincial Key Lab on Speech and Image Information Processing
E-mail: jiangdm@nwpu.edu.cn Tel: +86-29-88431532
Fengna Wang, Isabel Gonzalez, Enescu Valentin and Hichem Sahli
VUB-NPU Joint AVSP Research Lab
Vrije Universiteit Brussel, Brussels, Belgium
Electronics & Informatics Department
E-mail: hsahli@vub.ac.be Tel:+32-2-6292916

***Abstract*—Deep Neural Network Hidden Markov Models, or DNN-HMMs, are recently very promising acoustic models achieving good speech recognition results over Gaussian mixture model based HMMs (GMM-HMMs). In this paper, for emotion recognition from speech, we investigate DNN-HMMs with restricted Boltzmann Machine (RBM) based unsupervised pre-training, and DNN-HMMs with discriminative pre-training. Emotion recognition experiments are carried out on these two models on the *eNTERFACE'05* database and *Berlin* database, respectively, and results are compared with those from the GMM-HMMs, the shallow-NN-HMMs with two layers, as well as the Multi-layer Perceptrons HMMs (MLP-HMMs). Experimental results show that when the numbers of the hidden layers as well hidden units are properly set, the DNN could extend the labeling ability of GMM-HMM. Among all the models, the DNN-HMMs with discriminative pre-training obtain the best results. For example, for the *eNTERFACE'05* database, the recognition accuracy improves 12.22% from the DNN-HMMs with unsupervised pre-training, 11.67% from the GMM-HMMs, 10.56% from the MLP-HMMs, and even 17.22% from the shallow-NN-HMMs, respectively.**

I. Introduction

Speech emotion recognition aims at recognizing the underlying emotional state of the speaker from his or her speech signal. This is mainly motivated by intelligent Human - Machine Interaction required for different kinds of applications. In the field of speech emotion recognition, a number of classification approaches have already been explored. According to the used acoustic emotional features, the recognition models can be classified into two types: 1) for suprasegmental prosodic features, such as the mean, median, standard deviation, range, or percentile of short time pitch (energy), estimated over the whole utterance, global models such as Gaussian mixture model(GMM), support vector machine(SVM), artificial neutral networks (ANN) and k-NN have been adopted. 2) for frame based dynamic spectral features like Mel Filterbank Cepstrum Coefficient (MFCC), dynamical models such as Hidden Markov Model (HMM) are considered [1], [2]. Compared to the global models, dynamic modeling approaches provide a better consideration of the temporal dynamics of emotions. But at the same time, the limited representational capacity of HMMs prevents them from modeling streams of interacting knowledge sources in the speech signal which may require deeper architectures with multiple layers of representations [3]. Moreover, Gaussian mixture models(GMMs) in HMM have a serious shortcoming that they are statistically inefficient for modeling data that lie on or near a nonlinear manifold in the data space [4].

In recent years, a novel hybrid model architecture, Deep Neural Network - Hidden Markov Model (DNN-HMM), has been proposed and widely used in speech recognition [5], [6]. A deep neural network (DNN), which is able to capture the underlying nonlinear relationship among data, is the conventional multi-layer perceptrons with many layers, where training is typically initialized by a pre-training algorithm [4], [7]. For example, [5] proposed an unsupervised pre-training method to train a deep belief network, which has the strong ability of feature learning and provides a better recognition result. In [6], the authors further replaced the unsupervised pre-training method of [5] by a supervised pre-training method and claimed improved recognition results.

In the literature, several works have been dedicated to DNN-HMMs based large vocabulary continuous speech recognition. However, to our knowledge only few works on the application of DNN-HMMs in emotion recognition, have been reported. In [8], a Generalized Discriminant Analysis (GerDA) based on DNNs, is proposed to learn the discriminative features for classifying high or low of arousal and positive or negative valence.

In this work, we go beyond feature selection and follow the idea of [5] for continuous speech recognition, we propose exploring the application of DNN-HMM framework in speech

978-0-7695-5048-0/13 $26.00 © 2013 IEEE
DOI 10.1109/ACII.2013.58

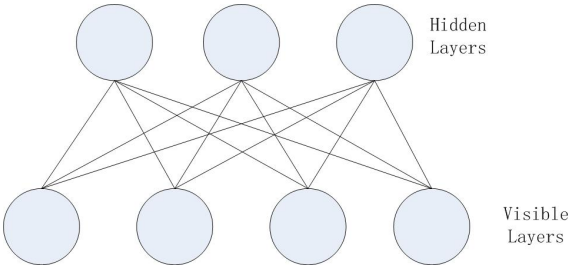

Fig. 1. Architecture of Restricted Boltzmann Machine, the connection between the units is symmetrical.

emotion recognition. In this approach the DNN can be viewed as a complex discriminative feature extractor extracting environment and speaker-independent representations optimized to predict the emotion class. Furthermore, we demonstrate that these models show improvement in emotion classification performance over baselines that do not employ deep learning. Moreover, we compare the DNN-HMMs with unsupervised and supervised pre-training methods, and illustrate our approach with experimental results on the *eNTERFACE'05* [9] and *Berlin* [10] speech emotion databases.

The remainder of this paper is organized as follows. In section II we briefly introduce the deep neural network, and outline the general pre-training strategies we used in this work. In section III, we describe the basic ideas of the proposed DNN-HMMs for speech emotion recognition along with the training and decoding strategies. In section IV experimental results using the *eNTERFACE'05* and the *Berlin* databases are discussed. Finally, Section V draws conclusions and outlines the future work.

II. Deep Neural Network

DNN is a feed-forward artificial neural network that has more than one hidden layers. Each hidden unit uses a nonlinear function to map the feature input from the layer below to the current unit. In our work, we use the traditional logistic function as the mapping function.

$$y = \frac{1}{1 + e^{-(b+xw)}} \tag{1}$$

where x denotes the input feature, w denotes the weights between connections, b denotes the bias and y denotes the output unit. DNN is capable of modeling very complex and highly nonlinear relationships between inputs and outputs, due to its flexible structure with multiple hidden layers and multiple hidden units.

DNN can be discriminatively trained by back-propagating (**BP**) derivatives of a cost function that measures the discrepancy between the target outputs and the actual outputs produced for each training case [6]. There are two methods to pre-train a Deep Neural Network, the unsupervised pre-training method [7], and the so called discriminative pre-training method, being a supervised pre-training approach [6].

A. Unsupervised pre-training

Unsupervised pre-training method uses stacked Restricted Boltzmann Machine (RBM) to initialize the Deep Neural Network. RBM is a type of undirected graphical model constructed from a layer of binary stochastic hidden units and a layer of stochastic visible units, which will either be Bernoulli or Gaussian distributed conditional on the hidden or visible units. Fig. 1 depicts the structure of a RBM. Since in our work, the input of the network are real values, we use a RBM in which the hidden units are Bernoulli distributed, and the visible units are linear real-valued variables with Gaussian noise [11].

A RBM assigns an energy to every configuration of visible and hidden state vectors, denoted by **v** and **h** respectively, according to [5]:

$$E(\mathbf{v}, \mathbf{h}) = \sum_i \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_j b_j h_j - \sum_{i,j} w_{ij} \frac{v_i}{\sigma_i} h_j, \tag{2}$$

where $\mathbf{w} = \{w_{ij}\}$ is the matrix of visible/hidden connection weights, b_j is the bias of the hidden unit h_j, a_i and σ_i are the mean and variance of the input variable v_i. When the visible unit is Gaussian distributed and the hidden unit is Bernoulli distributed, the conditional distribution (with $\sigma_i = 1$) are:

$$p(h_j = 1|\mathbf{v}) = \frac{1}{1 + exp(-b_j - \sum_i v_i w_{ij})} \tag{3}$$

$$p(v_i|\mathbf{h}) = \mathcal{N}(c_i + \sum_j h_j w_{ij}, 1), \tag{4}$$

where c_i is the bias of the visible unit a_i, $\mathcal{N}$ is the probability density function of normal distribution.

Before writing an expression for the log probability assigned by a RBM to some visible vector **v**, it is convenient to define a quantity known as the free energy [12]:

$$F(\mathbf{v}) = -log(\sum_h e^{-E(v,h)}) \tag{5}$$

Using $F(v)$, we can write the per-training-case log likelihood as

$$\xi(\theta) = -F(\mathbf{v}) - log(\sum_v e^{-F(v)}) \tag{6}$$

with θ denoting the model parameters. From the iteration t to $t+1$, the typical model parameter w_{ij} is updated as

$$\Delta w_{ij}(t+1) = m\Delta w_{ij}(t) - \epsilon \frac{\partial \xi(\theta)}{\partial w_{ij}} \tag{7}$$

where ϵ is the learning rate of RBM, m is the momentum factor used to smooth out the weight updates, and

$$-\frac{\partial \xi(\theta)}{\partial w_{ij}} = < v_i h_j >_{data} - < v_i h_j >_{recon} \tag{8}$$

where $< \cdot >_{data}$ is the expectation of the training data, and $< \cdot >_{recon}$ is the expectation w.r.t the distribution of the

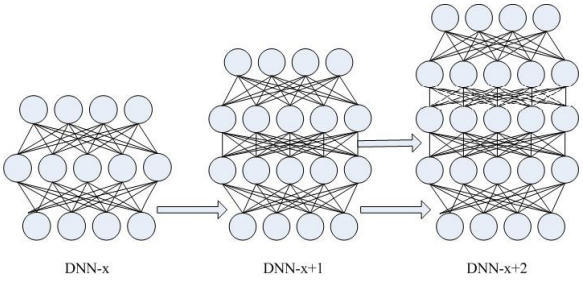

Fig. 2. Supervised pre-training method - using a network with less hidden layers to initialize a deeper network

reconstructed data. The learning rule of the bias of the hidden unit is:

$$\Delta b_j = \epsilon(< h_j >_{data} - < h_j >_{recon}) \quad (9)$$

Once the unsupervised pre-training is done, the obtained parameters are used as the initial values of the DNN, and BP is adopted to fine-tune the network.

B. Discriminative pre-training

To remedy the modeling inaccuracies in unsupervised pre-training, we follow the alternative proposed by [5] referred to as discriminative pre-training (DPT). The general architecture is shown in Fig. 2. It works as follows, in a first step, a layer-wise Back Propagation (BP) is used to train a one-hidden-layer DNN to full convergence using every frame's state label, then the *softmax* layer [5] is replaced by another randomly initialized hidden layer and a new random *softmax* layer on top, and the network is discriminatively trained again to full convergence. The process is repeated until the desired number of hidden layers is reached. As stated in [5], this is similar to a greedy layer-wise training [3], but differs in that of [3] only by the updates of newly added hidden layers. [5] showed that DPT outperforms DNNs with unsupervised pre-training and DNNs without pre-training.

III. DNN-HMM Based Emotion Recognition

In conventional GMM-HMMs based emotion recognition, the observation probabilities are modeled using GMMs under the maximum likelihood criterion. The potential of such model is restricted since GMMs are statistically inefficient for modeling data that lie on or near a nonlinear manifold in the data space [5]. To overcome this restriction, we propose a hybrid Deep Neural Network - Hidden Markov Model (DNN-HMM) for speech emotion recognition, where the output of the DNN are fed to the HMM as replacement of the GMMs.

A. Hidden Markov Models

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. A HMM, represented as $\lambda = (A, B, \pi)$, consists of the following elements:

1) The number of states in the model denoted as Q, the set of states denoted as $S = \{s_1, s_2, \ldots, s_Q\}$, and q_t the state at time t.
2) $A = \{a_{ij}\}$, the state transition probability distribution, with
$$a_{ij} = P(q_{t+1} = s_j | q_t = s_i), 1 \leq i, j, \leq Q \quad (10)$$
3) $B = \{b_i(o_t)\}$, the observation probabilities, where $b_i(o_t)$ represents the probability of observing o_t at state s_i. B is represented by a finite mixture:
$$b_i(o_t) = \sum_{m=1}^{M} c_{im} \aleph(o_t, \mu_{im}, \mathbf{U}_{im}), 1 \leq i \leq Q \quad (11)$$
where c_{im} is the mixture coefficient for the *m*th mixture in state s_i, and $\aleph$ any log-concave or elliptically symmetric density, with mean vector $\boldsymbol{\mu}_{im}$ and covariance matrix $\mathbf{U}_{im}$ for the *m*th mixture component in state i. When $\aleph$ is a Gaussian function, Equation 11 is a GMM, and the corresponding HMM is called a GMM-HMM.
4) $\pi = \{\pi_i\}$, the initial state distribution, where $\pi_i = P(q_1 = s_i)$, $1 \leq i \leq Q$

To be able to use the HMM, there are two problems that one should solve [13]:

- Learning problem: Given a set of ground truth $\mathbf{X}$ (referred to as *training set* in the following), the learning procedure is to find the set of model parameters $\lambda^* = \{A^*, B^*, \pi^*\}$ such that $\lambda^* = \arg\max_\lambda P(\mathbf{X}|\lambda)$, i.e., find the model parameters that better fit the training set. The forward-backward algorithm is used to calculate $P(\mathbf{X}|\lambda)$, while the Baum-Welch algorithm is employed to solve the learning problem [13].
- Decoding problem: Given a model λ and a sequence of new observations $O = (o_1, o_2, \ldots, o_T)$ (referred to as *testing set* in the following), the decoding procedure is defined as the problem of finding the hidden state sequence $(q_1, ..., q_T)$ that have most likely produced that observation. The solution of this problem is given by the Viterbi algorithm [13] as

$$P(O|\lambda) = \max_{q_1,\ldots,q_T} \pi_{q_1} \prod_{t=2}^{T} p(q_t|q_{t-1}) b_{q_t}(o_t) \quad (12)$$

In the case of speech emotion recognition, we train C HMMs $\{\lambda_c, (c = 1, \ldots, C)\}$ for C discrete emotion classes. For a new speech input O, it is assigned to the emotional class

$$c^* = \underset{1 \leq c \leq C}{argmax}\, P(O|\lambda_c) \quad (13)$$

with $P(O|\lambda_c)$ calculated from the Viterbi algorithm.

B. DNN-HMM

The key difference between DNN-HMM and GMM-HMM is the using of DNN (instead of GMM) to estimate the observation probabilities. We actually use the DNN to model $p(q_t|o_t)$, the posterior probability of the state given the observation vector o_t , which is possible since $p(q_t)$ is easy to estimate from an initial state-level alignment of the training set.

1) DNN-HMM Training Procedure: The detailed training process for emotion recognition is as follows:

a) For each emotion class $c(c = 1, \ldots, C)$, a left to right GMM-HMM λ_c with Q states is trained using the training speech sentences of class c.

b) For each speech sentence $O = (o_1, o_2, \ldots, o_T)$ in the training set c, the Viterbi algorithm of the GMM-HMM according to Equation 12, is performed on λ_c to obtain an optimal state sequence $(q_1^c, ..., q_T^c)$, and each state q_t^c is assigned a label $L_i(i \in (1, \ldots, C \times Q))$ according to a state-label mapping table.

c) All the training sentences, together with their labeled state sequences are used as inputs to train a DNN, whose outputs are the *posterior probabilities* of the $C \times Q$ output units. The training of the DNN is performed using BP algorithm with (i) the unsupervised pre-training, or (ii) the discriminative pre-training described in Section II.

2) DNN-HMM Recognition Procedure: In the emotion recognition process, for an input speech sentence $O = (o_1, o_2, \ldots, o_T)$, one should estimate the probability $p(O|\lambda_c)$ for each emotion class c, and get the final recognition result according to Equation 13. In GMM-HMM, this probability is obtained via the Viterbi algorithm with Equation 12.

In DNN-HMM, we adopt the following procedure to calculate the probability $p(O|\lambda_c)$.

a) The input feature sequence O is firstly input into the DNN, obtaining the posterior probabilities $\{p(L_i|o_t)\}_{i=1,\ldots,C\times Q}$ as outputs. Then the posterior probability $p(q_t = S_k^c|o_t)$ can be obtained from $p(L_i|o_t)$, by mapping the label L_i to the state k of the model c, using a state-label mapping table.

b) According to the Bayesian principle, we calculate the likelihood probability $p(o_t|q_t)$ as

$$p(o_t|q_t) = \frac{p(q_t|o_t)p(o_t)}{p(q_t)} \tag{14}$$

In our implementation, the prior probability of each state, $p(q_t)$, is calculated from (occurrences of) the training set, and $p(o_t)$ can be assigned a constant since the observation feature vectors are regarded as independent of each other.

c) For each emotion model λ_c, the Viterbi algorithm is performed to calculate the likelihood probability $p(O|\lambda_c)$ according to Equation 12. However, here the probability $b_{q_t}(o_t)$ is replaced by $p(o_t|q_t)$ calculated using Equation 14.

IV. Experimental Results and Analysis

A. Emotional Speech Databases and Emotion Features

In this study we perform emotion recognition experiments on the eNTERFACE'05 and Berlin emotion databases. eNTERFACE'05 contains 495 utterances in 6 emotions (anger, happiness, sadness, fear, surprise and disgust). Berlin database contains 493 utterances selected from the utterances of 10 professional actors (five males and five females) each speaking 10 sentences in 7 emotions (anger, boredom, disgust, fear, happiness, sadness and neutral). In each database, we randomly select 50% of the speech sentences as the training set, 10% as the developing set to find the optimal parameters of DNN such as the learning rate and momentum, and the rest 40% as the testing set. Both the training and testing partitions are speaker independent in the two databases.

For the speech emotion recognition, MFCC feature vectors of dimension 42 are extracted, including 14 MFCC coefficients and their first as well second order derivatives, with a Hamming window of 25ms and window shift of 10ms. In DNN based speech recognition, concatenated features of several adjacent frames are often adopted to improve the recognition performance [14]. Therefore, in our DNN-HMM modeling, we adopt a sliding hamming window of m frames (5 for the eNTERFACE'05 database, and 3 for the Berlin database) to weight the MFCC features, then concatenate the weighted MFCC features into a higher dimension feature vector. The sliding step of the Hamming window is 1 frame.

For each of the emotions in the database, the HTK toolkit [15] has been used to train a baseline left-right GMM-HMM with 5 states and 17 Gaussian mixtures. In the training process of DNN, we adopt a mini-batch approach on the training data. Each 100 concatenated MFCC feature vectors, together with their labeled states, are input as a batch to train the DNN, and the DNN parameters are updated by Equations 7 and 9 using the following batch of 100 concatenated MFCC feature vectors. After all the training data has been used to update the parameters, the connection weights of the trained DNN are compared to the weights of the last iteration. If the difference between the weights of successive iterations is lower than a given threshold, then the training process stops, otherwise the training process will continue until 30 iterations.

Once the training process finishes, following the step $a)$ of Section III-B2, the concatenated MFCC feature vectors of the developing set are input into the DNN to get the state labels of each frame. These states are then compared to the "ground-truth" states obtained by the Viterbi algorithm based on the GMM-HMM, and an error rate is calculated (details can be found in section IV-D). The DNN parameters, i.e. the learning rate, momentum and number of hidden layers, which get the low error rates on the developing set, are chosen as the candidate DNN parameters for emotion recognition experiments on the testing set.

B. Emotion Recognition on the eNTERFACE'05 Database

We perform speech emotion recognition on the DNN-HMMs with RBM based unsupervised pre-training and discriminative pre-training, respectively, and compare the results with those from the shallow-NN-HMM with discriminative pre-training in which the DNN has only two layers, and those from the Multi-Layer Perceptrons HMM (MLP-HMM) in which BP is used directly to train the DNN without pre-training [6]. Emotion recognition accuracies are listed in Table I. One can notice that the DNN-HMM with discriminative pre-training obtains the best result, which is 12.22% higher than that of the DNN-HMM with unsupervised pre-training, 11.67% higher than that of the GMM-HMM, 10.56% higher

than that of MLP-HMM, and even 17.22% higher than that of the shallow-NN-HMM with 2 hidden layers.

TABLE I
eNTERFACE'05 - EMOTION RECOGNITION RESULTS (%)

Method	Accuracy
GMM-HMM	42.22
Shallow-NN-HMM (2 hidden layer)	36.67
DNN-HMM (discriminative pre-training, 6 hidden layers)	53.89
DNN-HMM(unsupervised pre-training, 6 hidden layers)	41.67
MLP-HMM (back-propagation, 6 hidden layers)	43.33

The confusion matrix of the emotion recognition using DNN-HMM with discriminative pre-training, which gets the best recognition result, is shown in Table II. One can notice that sadness obtains 85% correction rate. Nevertheless, the correction rates of the other 5 emotions are relatively low, with the values ranging from 41.67% to 55%. However, the correction rates from DNN-HMMs are more coherent than those from GMM-HMMs as shown between brackets in Table II, where the correct rates range from 20% to 60% except for 83.33% for sadness. Moreover, for most of the emotions having low correct rates in GMM-HMM, the recognition performance can be efficiently improved by DNN-HMM.

C. Emotion Recognition on the Berlin Database

The emotion recognition results on the Berlin database are summarized in Table III. One can notice that the DNN-HMM of 5 hidden layers with discriminative pre-training obtains the highest accuracy up to 77.92%, which is 3.64% higher than that of DNN-HMM with unsupervised pre-training, 7.98% higher than that of DNN-HMM without pre-training, and 1.74% higher than that of the traditional GMM-HMM.

D. Analysis of the Numbers of Hidden Layers and Hidden Units

In order to further explore the influence of the numbers of hidden layers as well as hidden units, we have carried out DNN-HMM based emotion recognition experiments using DNNs with different numbers of hidden layers.

Fig. 3 shows the curve of the recognition accuracies versus the number of hidden layers in DNN-HMMs for the eNTERFACE'05 database. One can notice that the recognition accuracies increase with more hidden layers. When there are 6 hidden layers, the DNN-HMMs get the best performance. However, when the number of hidden layers is further increased to 7, the emotion recognition accuracy decreases.

Further, we evaluated the states which are output by the DNN-HMM, as well as checked the fitting extent of the DNN-HMM with respect to the GMM-HMM. For this analysis, after the DNN has been trained with discriminative pre-training by the training sentences of the database, the speech sentences from the developing set, along with their state labels obtained by the Viterbi algorithm based on GMM-HMM, were fed to the trained DNN to obtain the output states, which were then compared to the labeled states of corresponding frames. An error rate, defined as the number of states that are not correctly

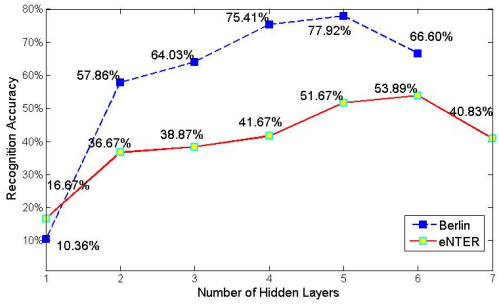

Fig. 3. Recognition Accuracy v.s. DNN's number of hidden layers

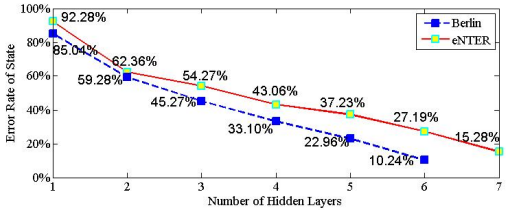

Fig. 4. Error Rate of states v.s. DNN's number of hidden layers

output by the DNN, divided by the number of states in all testing sentences, has been used as measure of robustness. To ensure correctness of this measure, we repeated the analysis 3 times by selecting different developing sets. The final error rate is calculated as the average of the 3 error rates.

The red curve in Fig. 4 shows the error rates versus the number of hidden layers on the eNTERFACE'05 database. One can notice that with increasing number of hidden layers, the error rates decreases rapidly. In other words, the more hidden layers, the more closely DNN-HMM fits the GMM-HMM. However, this does not mean that the DNN-HMM with more hidden layers would provides better emotion recognition performance. From Fig. 3, one can notice that the DNN-HMM with 6 hidden layers obtains 53.89% accuracy, while the one with 7 hidden layers obtains only 40.83%. This confirms that the DNN-HMM could indeed extend the ability of GMM-HMM. While the states are labeled from GMM-HMM, they are obtained via the Viterbi algorithm based on GMM-HMM which is trained using the Maximum Likelihood Estimation (MLE) method, some states could be unavoidably falsely labeled. However, DNN-HMM with proper number of hidden layers could improve the state labeling and hence obtains better emotion recognition results. Similar conclusion could be drawn for the Berlin database from Fig. 3 and Fig. 4.

In order to evaluate the number of hidden units in DNN, we list in Table IV the recognition accuracy versus number of hidden units, for the eNTERFACE'05. One can notice that the DNNs with 512 hidden units obtain the best results. When the hidden units are further increased to 1k, the recognition accuracy decreases, which means that proper number of hidden units should also be properly set to get the best result. The same results can be obtained for the Berlin database.

TABLE II
eNTERFACE'05 - DNN-HMM v.s. (GMM-HMM) Confusion Matrix (%)

	anger	sadness	happiness	disgust	fear	surprise
angry	55.00 (20.00)	6.67 (13.33)	5.00 (20.00)	11.67 (0.00)	11.67 (36.67)	10.00 (10.00)
sad	0.00 (0.00)	85.00 (83.33)	0.00 (0.00)	5.00 (6.67)	6.67 (10.00)	3.33 (0.00)
happy	8.33 (0.00)	6.67 (13.33)	46.67 (40.00)	16.67 (6.67)	11.67 (30.00)	10.00 (10.00)
disgust	10.00 (13.33)	10.00 (13.33)	3.33 (6.67)	43.33 (20.00)	20.00 (23.33)	13.33 (23.33)
fear	11.67 (0.00)	20.00 (23.33)	6.67 (0.00)	6.67 (6.67)	41.67 (60.00)	13.33 (10.00)
surprise	18.33 (6.67)	11.67 (23.33)	1.67 (3.33)	6.67 (3.33)	10.00 (33.33)	51.67 (30.00)

TABLE III
Berlin - Emotion recognition results (%)

mthod	accuracy
GMM-HMM	76.18
Shallow-NN-HMM (2 hidden layers)	57.86
DNN-HMM (discriminative pre-training, 5 hidden layers)	77.92
DNN-HMM(unsupervised pre-training, 5 hidden layers)	74.28
MLP-HMM (back-propagation, 5 hidden layers)	69.94

TABLE IV
eNTERFACE'05 - Recognition accuracy (%) v.s. number of hidden units

Hidden Layers	Hidden Units	Accuracy
6	256	52.62
6	512	53.89
6	1k	47.55

V. Conclusion and Future Work

In this work, we propose Deep Neural Network Hidden Markov Models(DNN-HMMs) based emotion recognition from speech. Emotion recognition experiments are carried out on the eNTERFACE'05 database and Berlin database, on the DNN-HMMs with RBM based unsupervised pre-training, or with discriminative pre-training, respectively. Results are compared with those from the GMM-HMMs, the shallow-NN-HMMs, and the Multi-layer Perceptrons HMMs (MLP-HMMs). Experimental results show that when the numbers of the hidden layers as well hidden units are properly set, the DNN-HMM could extend the labeling ability of GMM-HMM. Therefore among all the models, the DNN-HMMs with discriminative pre-training obtain the best results. In our future work, we will consider evaluating this approach for Facial Action Unit recognition, as well audio-visual emotion recognition, to improve the emotion recognition accuracy.

VI. ACKNOWLEDGMENT

This work is supported within the framework of the national natural science foundation project of China (grant 61273265), the Shaanxi provincial key international cooperation project (2011KW-04), the LIAMA-CAVSA project, the EU FP7 project ALIZ-E (grant 248116), and the VUB IRP5 project.

References

[1] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[2] T. L. Nwe, S. W. Foo, and L. C. De Silva, "Speech emotion recognition using hidden markov models," *Speech communication*, vol. 41, no. 4, pp. 603–623, 2003.

[3] A.-r. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[4] 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.

[5] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[6] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 24–29.

[7] 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.

[8] A. Stuhlsatz, C. Meyer, F. Eyben, T. ZieIke, G. Meier, and B. Schuller, "Deep neural networks for acoustic emotion recognition: raising the benchmarks," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5688–5691.

[9] O. Martin, I. Kotsia, B. Macq, and I. Pitas, "The enterface05 audio-visual emotion database," in *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*. IEEE, 2006, pp. 8–8.

[10] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of german emotional speech," in *Proc. Interspeech*, vol. 2005, 2005.

[11] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," vol. 19. MIT; 1998, 2007, p. 1345.

[12] J. Nicholson, K. Takahashi, and R. Nakatsu, "Emotion recognition in speech using neural networks," *Neural Computing & Applications*, vol. 9, no. 4, pp. 290–296, 2000.

[13] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[14] P. Schwarz, P. Matejka, and J. Cernocky, "Hierarchical structures of neural networks for phoneme recognition," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[15] 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," *Cambridge University Engineering Department*, vol. 3, 2002.