International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.6, No.6 (2013), pp.375-386
http://dx.doi.org/10.14257/ijsip.2013.6.6.34

Improved Bottleneck Feature using Hierarchical Deep Belief Networks for Keyword Spotting in Continues Speech

Yi Wang, Jun-an Yang and Hui Liu

Laboratory 404 of Electronic Engineering Institute, 460 Huangshan Road, Hefei 230037, China
Key Laboratory of Electronic Restriction of Anhui province, 460 Huangshan Road, Hefei 230037, China
wygggg@126.com, yangjunan@ustc.edu, christ592604@yahoo.com.cn

Abstract

Bottleneck (BN) feature has attracted considerable attentions by its capacity of improving the accuracies in speech recognition tasks. Recently, researchers have proposed some modified approaches for extracting more effective BN feature, but these approaches still need further improvement. In this paper, motivated by both deep belief networks (DBN) and hierarchical Multilayer Perceptron (MLP), we propose hierarchical DBNs based BN feature and employed it for keyword spotting task. The hierarchical DBNs based BN feature is constructed with two DBNs in series which are sequentially trained. The first DBN outputs the posterior probabilities features, as well as the second DBN transforms the posterior probability features into a low dimensional representation with the information pertinent to classification through the BN layer. Experiments on hierarchical DBNs based BN feature is conducted with TIMIT dataset and using Point Process Model as the baseline system. Experimental results show that the hierarchical DBNs based BN feature is more robust and can achieve better accuracies than other features.

Keywords: *Bottleneck Feature, Hierarchical Deep Belief Network, Keyword Spotting, Point Process Model*

1. Introduction

Compared with the large vocabulary continuous speech recognition (LVCSR) technology, keyword spotting has the advantages such as insensitive to circumstance change; less system resources requirement and faster recognition speed in detecting certain desired words in continue speech. Hence it has been widely used in audio indexing and speech data mining applications.

The recent research hotspot and difficulty in keyword spotting is focus on developing new features [1]; the reason is the state-of-the-art speech features such as Mel-frequency Cepstral Coefficients (MFCCs) or Perceptive Linear Predictive (PLP) are sensitive to noise on the one hand and have poor classifying capability on the other [2]. Hence aiming at overcoming the inherent deficiencies of these features, some researchers employ bottleneck (BN) feature which is based on Deep Belief Network (DBN) [3].

As the name suggest, DBN based BN features is constructed with DBNs in contrast with the conventional BN feature which is generated by multi-layer perceptron (MLP). It can be considered as a dimensionality reduction and nonlinear feature extraction technique, and the goal of BN feature is to derive a set of features with low dimension and higher classification

ISSN: 2005-4254 IJSIP

accuracy [3-5]. With the help of the pre-training procedure in DBN training [6, 7], DBN based BN feature can conquer the inherent flaws of MLP which is often get stuck in poor local optimum and has shown absolute improvements over the other features including the MLP based BN feature.

But the DBN based BN feature still needs further improvement. Firstly, the DBN based BN feature places a BN layer in the middle of the DBN which will degrade the frame accuracy at the output targets. Secondly, some researchers report that the probabilistic features can achieve same performances or, in some cases, outperform the classical features [8]. Last but not the least, the hierarchical MLPs, which constructed with two MLPs in series, has been successfully used in both probabilistic feature extraction and phonetic class conditional probabilities estimating. It has been proved that the hierarchical MLPs based system is more powerful than the single MLP based one [9]. Consider DBN is essentially an enhanced version of MLP, we believe that using DBN to take the place of MLP and build hierarchical DBNs will improve the performance of conventional BN feature, and achieve our goal of getting higher accuracy in keyword spotting.

In this paper, we propose hierarchical DBNs based BN feature for keyword spotting in continues speech. The hierarchical DBNs based BN feature is constructed with two DBNs in series which are sequentially trained. The first DBN uses long-term raw feature (e.g., 3-frame concatenating features) as input, and output the posterior probabilities feature. In the second DBN, the posterior probability feature estimates by the first DBN is transformed into a low-dimensional representation with the information pertinent to classification through the BN layer. Experiments are conducted with TIMIT database and using a Point Process Model which is a novel keyword spotting paradigm as the baseline system. The results show that the BN feature based on hierarchical DBNs can get better accuracies than other features.

The rest of this paper is organized as follows: In Section 2, we describe DBN and hierarchical DBNs based BN feature system. Section 3 analyzes the operating principle of point process model. Experimental results are provided and discussed in Section 4. Section 5 concludes the paper and discusses future work.

2. Hierarchical DBNs based BN Feature

2.1. DBN [6, 7, 10]

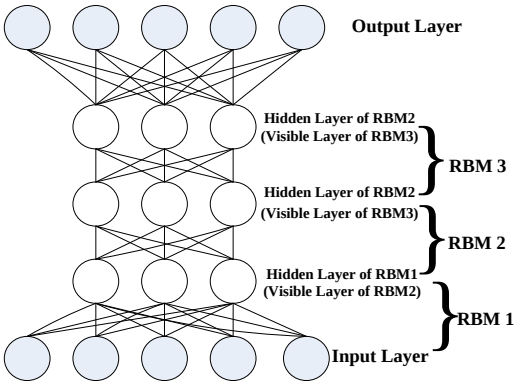

Figure 1. Schematic Representation of a DBN

DBN can be viewed as a composition of simple learning modules via stacking them, and this simple learning module is called Restricted Boltzmann Machines (RBMs). The schematic representation of a DBN is shown in Figure 1. A RBM is structured with two layers of

neurons, one layer of (typically Bernoulli) stochastic hidden units and one layer of (typically Bernoulli or Gaussian) stochastic visible units, where all visible units are connected to all hidden units, but there is no connection between the same layer. The joint distribution $p(\mathbf{v},\mathbf{h};\theta)$ of a RBM over the visible units $\mathbf{v}$ and hidden units $\mathbf{h}$, given the model parameters θ, is defined in terms of an energy function $E(\mathbf{v},\mathbf{h};\theta)$ of

$$p(\mathbf{v},\mathbf{h};\theta)=\frac{exp(-E(\mathbf{v},\mathbf{h};\theta))}{Z} \tag{1}$$

Where $Z=\Sigma_{\mathbf{v}}\Sigma_{\mathbf{h}}exp(-E(\mathbf{v},\mathbf{h};\theta))$ is a normalization factor. And for a Bernoulli (visible)-Bernoulli (hidden) RBM, the energy function can be defined as

$$E(\mathbf{v},\mathbf{h};\theta)=-\sum_{i=1}^{I}\sum_{j=1}^{J}w_{ij}v_ih_j-\sum_{i=1}^{I}b_iv_i-\sum_{j=1}^{J}a_jh_j \tag{2}$$

Where w_{ij} represents the symmetric interaction term between visible unit v_i and hidden unit h_j, b_i and a_j are the bias terms. But the Bernoulli-Bernoulli RBM is inconvenient for modeling real-valued data such as speech, so we adopt a Gaussian-Bernoulli RBM, and the energy function become

$$E(\mathbf{v},\mathbf{h};\theta)=-\sum_{i=1}^{I}\sum_{j=1}^{J}\frac{w_{ij}v_ih_j}{\sigma_i}-\sum_{i=1}^{I}\frac{(v_i-b_i)^2}{2\sigma_i^2}-\sum_{j=1}^{J}a_jh_j \tag{3}$$

Notice that each visible unit v_i adds a parabolic (quadratic) offset to the energy function, where σ_i controls the width of the parabola. Then the conditional probabilities can be efficiently calculated as

$$p(h_j=1\mid\mathbf{v};\theta)=1/(1+e^{-(\sum_{i=1}^{I}\frac{w_{ij}v_i}{\sigma_i}+a_i)}) \tag{4}$$

$$p(v_i=1\mid\mathbf{h};\theta)=1/(1+e^{-(\sum_{j=1}^{J}w_{ij}h_j+b_j)}) \tag{5}$$

Taking the gradient of the log likelihood $\log p(\mathbf{v};\theta)$, we can derive the update rule for the RBM weights as

$$\Delta w_{ij}=<E_{data}(v_ih_j)>-<E_{model}(v_ih_j)> \tag{6}$$

Where $<E_{data}(v_ih_j)>$ and $<E_{model}(v_ih_j)>$ are the expectation observed in the training set and the model respectively. In practical use, we employ an algorithm called Contrastive Divergence (CD) to approximation the gradient of $<E_{model}(v_ih_j)>$ where $<E_{model}(v_ih_j)>$ is replaced by running the Gibbs initialized at the data for one full step.

As we discussed above, a DBN is built up by stacking a number of RBMs layer by layer from bottom up. This efficient layer-by-layer greedy learning strategy is called unsupervised pre-training, theoretical justification given in [7, 11] show that this procedure can improve a variational lower bound on the likelihood of the training data under the composite model. After the unsupervised pre-training, we still need a supervised procedure by fine-tuning the resulting weights using gradient descent learning to improve the performances of DBN. The main idea behind DBN is using unsupervised procedure to set the weights of the network to be closer to a good solution than random initialization, thus avoiding local minima.

2.2. Hierarchical DBNs based BN Feature

Hierarchical DBNs based BN feature is constructed with two DBNs in series. As shown in Figure 2, the first DBN is pre-trained and fine-tuned to minimize the cross-entropy between the hypothesized class probabilities and the targets.

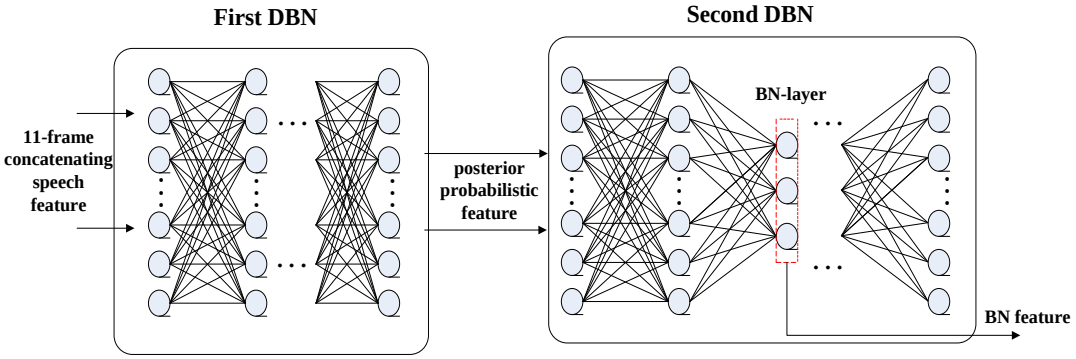

Figure 2. Structure of Hierarchical DBNs based BN Features

Training of the first DBN is very similar to DBN trainings done for other speech recognition applications [3, 10]; the only difference is the input feature we use in this paper are 11-frame concatenating features, which are constructed by augmenting the current speech frame with its neighboring 3 frames within a context window (1+1+1). The reason is according to the latest study on DBN that the gains of DBN are mostly attributed to the feature vectors that are concatenated from a long temporal context [12]. Additionally, the feature for each frame are 43-dimensional conventional speech feature, which is the combinations of 39-dimensional MFCC feature (static, first and second derivatives) and 4-dimensional pitch feature. Hence the dimension of the input feature is 129(43×3) and the number of nodes of the input layer is configured to be the same. Then the structure of the first DBN can be displayed as "129-[2048-2048-2048-2048]-129" for example, figures in [] is the number of nodes of the 4 hidden layers.

The second DBN is topologically similar to the DBN based BN features, and the posterior probability features estimated by the first DBN are used as the input feature of the second DBN.

As the name suggests, BN feature is generated from the second DBN which one of the hidden layers (conventionally the middle layer) has a very small number of hidden unites relative to the other layers. This layer is described as BN layer. The benefits of BN feature are obvious that it not only embedded the input features with classification information which derived from the supervision, but also forced input feature into a low dimensional representation. The structure of the second DBN can be displayed as "129-[2048-1048-43-1048-2048]-129" for example. Particularly, it would be specially mentioned that when the training of the second DBN is done, the layers after BN layer should be discarded; and the output feature is generated from the BN layer.

Compared with other speech features, hierarchical DBNs based BN feature has the following advantages: firstly, they do not require strong assumptions on data distributaries which make them can simply concatenate different distributaries features together[13, 14]; secondly, when trained on large amount of data, DBN is invariant to speaker characteristics and environment specific information such as noise [13]; thirdly, DBN can be trained efficiently and are scalable with large amount of data[13]; fourthly,

the posterior probabilistic features produced by the first DBN have lesser nonlinguistic variabilities compared to classical spectral features, which make hierarchical DBNs can yield higher accuracies to single DBN based system[8]. Last but not the least, the proposed BN features is low-dimensional and embedded with classification information, which will make them more suitable for keyword spotting task.

3. Point Process Model

Point process model is a novel keyword spotting approach which operates within the sliding model. In recent years, it has attracted extensive attentions in the community of speech recognition and cognitive science. Point process model is built on the hypothesis that the linguistic contents underlying human speech is coded in event-based point process, and the hypothesis is supported by several strands of researches in the fields of linguistics and neuroscience [15-18]. The principles of point process model can be simply divided into two parts: first is using detectors to generate an efficiently point process representation, which encoding the underlying linguistic contents. The detectors are phoneme classifiers essentially; second is building a suitable point process statistical model using the point process representation, and the model can distinguish keywords from the background utterances efficiently. Experimental results in [15, 16] have already proved that point process model has the capacity to generalize from a relatively small numbers of training examples and avoid the local optima of HMMs, the accuracy levels are comparable with other keyword spotting systems.

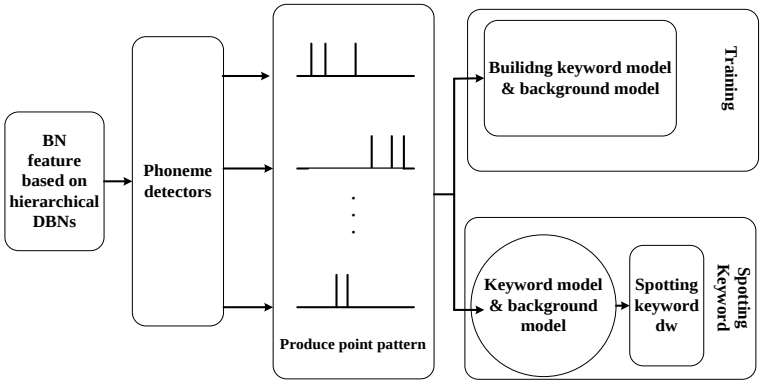

Figure 3. Framework of Point Process Keyword Spotting System Model

The framework of point process model is shown in Figure 3, and the training procedure of point process model can be summarized as follows:

Step 1: Build Gaussian Mixture Model (GMM) based phoneme detectors using 48 different phonemes examples, which are selected from the input hierarchical DBNs based BN feature according to [19].

Step 2: Map the input BN features to a collection of point patterns. The features input to the detectors take high values when the phoneme is expressed and take low values otherwise. Then mapping the input features to a time series, and by giving a threshold δ_p, we can compute the point process N_p for the specific phoneme p according to

$$N_p = \{t_i \mid p_{ip} > \delta_p \quad , \quad p_{ip} > p_{i\pm1,p}\}, i = 1 \cdots n \tag{7}$$

Where p_{ip} is the detect probability for the ith frame for phoneme p. Figure 4 show the point process representation of the keyword "greasy" using Eq. (6).

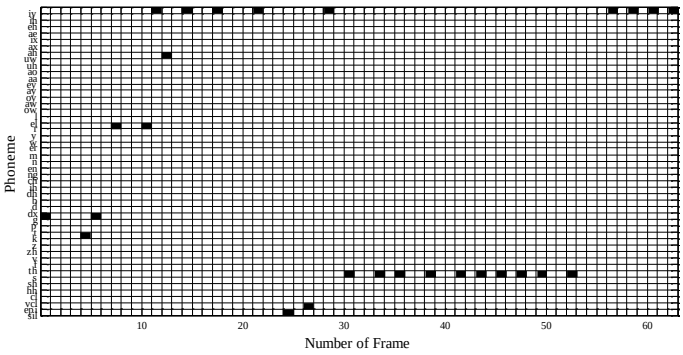

Figure 4. Point Process Representation of the Keyword "greasy"

Step 3: After obtaining the point process N_p of phoneme p, point process model builds an inhomogeneous Poisson process for keywords and a homogeneous Poisson processes for background utterances respectively. The reason is that we regard the probability distribution of the phoneme point process as the event streams in Queuing theory and use a homogeneous Poisson process to establish background model which is related with time interval, otherwise an inhomogeneous Poisson process models for each keyword which is unable to determine a specific time.

For a specific phoneme p and its point process set N_p, suppose that the point set obey the homogeneous Poisson process of λ_p, $\eta_p(t)$ represents the number of phoneme points in the time interval $(0,t]$. We can have the probability of k phoneme points occurred in $(t_a,t_b]$ according to the distribution of homogeneous Poisson process is

$$P_{a,b}(k) = P[\eta_p(t_b) - \eta_p(t_a) = k] = \frac{(\lambda_p \tau)^k e^{-\lambda_p \tau}}{k!}, \quad \tau = t_b - t_a \tag{8}$$

When τ is very small and we can lead to a corresponding density function $f(t) = \lambda_p e^{-\lambda_p t}$, then the likelihood of the point pattern becomes

$$P = \prod_{p \in P} (\lambda_p)^{n_p} e^{-\lambda_p T} \tag{9}$$

Training homogeneous Poisson process amounts to estimate λ_p for each phoneme p. Given N normalized-length training segments and the total number K of landmarks of phoneme p in those segments, the maximum-likelihood estimate is given by

$$\hat{\lambda}_p = \arg\max_{\hat{\lambda}} K_p \log \lambda - \lambda NT = \frac{K_p}{NT} \tag{10}$$

The training procedure of inhomogeneous Poisson processes for keywords is very similar to the homogeneous one. The difference between them is the λ_p now varies as a function of time. We solve this problem by factoring the inhomogeneous Poisson processes into D independent homogeneous processes operating in each division. Suppose $\lambda_{p,d}$ is the intensity function of phoneme p in the d division, then the likelihood of total point pattern becomes

$$P = \prod_{p \in P} \prod_{d=1}^{D} (\lambda_{p,d})^{n_{p,d}} e^{-\lambda_{p,d} \Delta T} \tag{11}$$

After training the background and keywords models respectively, we can construct the

detector function $d_w(t)$ in terms of the (log) likelihood ratio for spotting keyword

$$d_w(t) = \log[\sum_T \frac{P(\mathbf{O}(t) \mid T, \theta_k) P(T \mid \theta_k) \Delta}{T^{|\mathbf{O}_t|} P(\mathbf{O}(t) \mid T, \theta_b)}] \quad (12)$$

Where **O** is the set of observations in the utterance, θ_k and θ_b are the indicator function when keyword and background utterances presented respectively, T account for the duration for a particular keyword, interval Δ is the sliding windows durations, and the probability functions P have been calculated during the training procedure. When $d_w(t)$ over a threshold δ_w, then we can consider the keyword is expressed in speech.

4. Experiments and Results

4.1. Dataset

The efficacy of the hierarchical DBNs based BN feature is evaluated by performing keyword spotting experiments on TIMIT speech corpora [29].The TIMIT database consists of 4.3 hours (including 1.1 hours of NIST complete test set)of read speech. The training set has 4620 sentences collected from 462 speakers while testing set has 1620 sentences collected from 162 speakers, and there is no same speaker between the two sets. We analyze the input speech using a 25-ms Hamming window with 10-ms between the left edges of successive frames. And the features for each frame are 43-dimensional as we discussed in 2.2. The data were normalized to have zero mean and unit variance over the entire corpus.

4.2. Computational Setup

Training DBNs is quite computationally expensive as we mentioned in 2.1; therefore we accelerate the training procedure by exploiting a graphics processor (GPU). The experimental results show that the training speed of a single GPU is 20 times than an Intel 2.66-GHz Xeon mononuclear, which greatly saves time in DBN training.

4.3. Experimental Setup

Three different experiments are designed to verify the validity of the hierarchical DBNs based BN feature. Experiment 1 compares our hierarchical DBNs based BN feature with original MFCC, single MLP, single DBN and hierarchical MLPs based BN feature to verify the novel BN feature is able to increase the keyword spotting accuracies.

Aiming at exploiting the relationship between keyword spotting accuracies and the number of hidden layers as while as the position of the BN layer in second DBN, experiment 2 compares our hierarchical DBNs with different hidden layers and position of the BN layer in second DBN.

Similarly, experiment 3 compares the hierarchical DBNs based BN feature with different BN layer size in second DBN in order to find out the best size of BN layer.

4.4. Experimental Results

4.4.1. Experiment 1

In experiment 1, we adopt a 6 layer (with 4 hidden layer) DBN to build up the first DBN of the hierarchical DBNs, then the structure of the first DBN can be displayer as 129- [2048-2048-2048-2048]-129; the second DBN employs a 7 layer (with 5 hidden layer) DBN which the topological structure is 129-[2048-1048-43-1048- 2048]-129. For comparison, the

structure of the single MLP and single DBN is set to same as the second DBN of our hierarchical DBN. The hierarchical MLPs are set to just the same as the hierarchical DBNs. The number of Gaussian component in GMM based point process model is set to 8. All 4620 sentences in TIMIT training database are used in this experiment. Table 1 shows the Figure of Merits (FOM) of the five different features in keyword spotting.

Table 1. Performance Comparison between Hierarchical DBNs, Original MFCC, Single MLP, Hierarchical MLPs, and Single DBN based BN Feature

Features	FOM (%)
Original MFCC	92.17
Single MLP based BN features	92.66
Hierarchical MLPs based BN features	93.87
Single DBN based BN features	93.17
Hierarchical DBNs based BN features	95.55

The results in Table 1 demonstrate that the FOM of hierarchical DBN based BN feature is at least 2% better than the other features.

4.4.2. Experiment 2

In order to exploit the relationship between keyword spotting accuracies and the different structures of hierarchical DBNs based BN feature. We construct the second DBN with 5, 7 and 9 hidden layers as while as the BN layer is placed at the middle, in front and in back of the middle layer (e.g. for a second DBN with 5 hidden layer, we choose the second, third and fourth hidden layer as the BN layer respectively for experiments). The different structures of hierarchical DBNs are displayed as 129-[2048-1048-43-1048- 2048]-129 in Table 2 below. Other configurations of Experiment 2 are set to same as Experiment 1.

Table 2. Performance Comparison of Different Structures of Hierarchical DBNs based BN Feature

Different second DBN structures	FOM (%)
129-[2048-43-1048-1048- 2048]-129	93.67
129-[2048-1048-43-1048- 2048]-129	95.55
129-[2048-1048-1048-43- 2048]-129	94.83
129-[2048-1048-43-512-512-1048-2048]-129	91.31
129-[2048-1048-512-43-512-1048-2048]-129	93.21
129-[2048-1048-512-512-43-1048-2048]-129	93.12
129-[2048-1048-512-43-256-256-512-1048- 2048]-129	91.01
129-[2048-1048-512-256-43-256-512-1048- 2048]-129	92.14
129-[2048-1048-512-256-256-43-512-1048- 2048]-129	92.79

The results listed in Table 2 show that the second DBN with 5 hidden layers performs better than other structures. We believe that when DBN get "deeper", it not only taking longer training time, but also leading to the BN layer get less classification information from supervision. Moreover, the BN layer which locates in the middle of the DBN gets high FOM than in other positions. It suggests us that symmetric structures maybe the best topological structure of the second DBN.

4.4.3. Experiment 3

Finally, we conduct Experiment 3 to find out the best size for BN layer. Configurations of Experiment 3 are set to same as Experiment 1; the only difference is the BN layer size is ranging from 20 to 60. The result of Experiment 3 is shown in Table 3

Table 3. Performance Comparison of Different Size of BN Layer of Hierarchical DBNs based BN Feature

BN layer size	FOM (%)
20	95.23
30	95.32
43	95.55
50	95.63
60	95.17

From Table 3, we can see that FOM is not sensitive to the size of BN layer. And the size can be set to be same as the dimensionality of the original speech frames.

5. Conclusion

In this paper, we propose hierarchical DBNs based BN feature, and use this novel feature for keyword spotting task. The hierarchical DBNs are constructed with two DBNs in series which can combine the advantages of both DBN and hierarchical architecture. Experimental results on TIMIT dataset show that the hierarchical DBNs based BN features can yield 2% improvement compared with other features including single DBN based BN features. Furthermore, we also investigate the performances of our hierarchical DBN based BN feature with different DBN structures and size of the BN layer, results show that the second DBN with 5 hidden layers and 43 neurons of the BN layer which located in the middle of the whole structure achieve the best performances.

As we discussed above, training DBN is quite computationally expensive, and we also find there are only around 5% neurons are active together at a given time in training. Hence in the future work, we will focus on introducing the notion of sparse distributed representation and build deep architectures with sparse representations in order to improve the efficiency of DBN training.

Acknowledgements

The authors would like to thank Dr. Guo-ping Hu, Dr. Si Wei and Mr. Jia Pan at Anhui USTC iFlytek Corporation for preparing the dataset, and engaging in valuable discussions for DBN. This paper is also supported by national natural science foundation of China under Grant No. 61272333 and Anhui Provincial Natural Science Foundation under Grant No.1208085MF94, No.1308085QF99.

References

[1] M. Picheny, D. Nahamoo, V. Goel, B. Kingsbury, B. Ramabhadran and S. J. Rennie, "Trends and Advances in Speech Recognition", IBM Journal of Research and Development, vol. 55, no. 5, **(2011)**, pp. 2: 1-2: 18.
[2] H. Yang, S. Sharma, S. van Vuuren and H. Hermansky, "Relevance of time-frequency features for phonetic and speaker-channel classification", Speech Communication, vol. 31, no. 1, **(2000)**, pp. 35-50.
[3] D. Yu and M. Seltzer, "Improved Bottleneck Features Using Pre-trained Deep Neural Networks," In INTERSPEECH 2011 Proceedings, Florence, **(2011)** August.
[4] F. Grézl, M. Karafiat, S. Kontar and J. Cernock, "Probabilistic and Bottle-neck Features for LVCSR of meetings," IEEE Int. Conf. on Acoustics, Speech, and Signal Processing Proceedings, Honolulu, **(2007)** April.
[5] F. Grézl, M. Karafiat and M. Janda, "Study of Probabilistic and Bottle-Neck Features in Multilingual Environment", Automatic Speech Recognition and Understanding (ASRU) Proceedings, Honolulu, **(2011)** December.
[6] G. Hinton, S. Osindero and Y. Teh , "A Fast Learning Algorithm for Deep Belief Nets," Neural Computation, vol. 18, no. 7, **(2006)**, pp. 1527-1554.
[7] G. Hinton and R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," Science, vol. 312, no. 5786, **(2006)**, pp. 504-507.
[8] C. Plahl, R. Schlüter and H. Ney, "Hierarchical Bottle Neck Features for LVCSR," in Proceeding of INTERSPEECH, Makuhari, **(2010)**, pp. 1197-1200.
[9] G. Sivaram, H. Hermansky, "Sparse Multilayer Perceptron for Phoneme Recognition," IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 1, **(2012)**, pp. 23-29.
[10] L. Deng, "An Overview of Deep-Structured Learning for Information Processing," In Asian-Pacific Signal and Information Processing-Annual Summit and Conference Proceeding, Xian, **(2011)**, July.
[11] A. Mohamed, G. Hinton, G. Penn, "Understanding how deep belief networks perform acoustic modeling". in Proceeding of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Kyoto, **(2012)** March.
[12] J. Pan, C. Liu, Z. Wang, Y. Hu, H. Jiang, "Investigation of Deep Neural Networks (DNN) for Large Vocabulary Continuous Speech Recognition: Why DNN Surpasses GMMs in Acoustic Modeling," in Proceeding of IEEE Int. Conf. on Chinese Spoken Language Processing, **(2012)**, unpublished.
[13] A. Mohamed, T. Sainath, G. Dahl, B. Ramabhadran, G. Hinton and M. Picheny, "Deep Belief Networks Using Discriminative Features for Phone Recognition," IEEE Int. Conf. on Acoustics, Speech, and Signal Processing Proceedings, Prague, **(2011)** May.
[14] S. Ikbal. "Nonlinear Feature Transformations for Noise Robust Speech Recognition," Ph.D. dissertation. ´ Ecole Polytechnique F´ ed´ erale de Lausanne (EPFL), Switzerland, **(2004)**.
[15] A. Jansen and P. Niyogi, "Point Process Models for Spotting Keywords in Continuous Speech," IEEE Transaction on Audio, Speech, and Language Processing, vol. 17, no. 8, **(2009)**, pp. 1457-1470.
[16] A. Jansen. "Point Process Models for Event-Based Speech Recognition," Speech Communication, vol. 51, no. 12, **(2009)**, pp. 1155-1168.
[17] K. N. Stevens, Acoustic Phoenetics. Cambridge: MIT Press, **(1998)**.
[18] N. Suga, "Basic acoustic patterns and neural mechanisms shared by humans and animals for auditory perception", In: Listening to Speech: An Auditory Perspective, Greenberg S and Ainsworth WA, Eds.Mahwah, NJ: Lawrence Erlbaum Associates, **(2006)**.
[19] K. Lee and H. Hon, "Speaker-Independent Phone Recognition Using Hidden Markov Models," IEEE Transactions on Audio, Speech, and Language Processing, vol. 37, no. 11, **(1989)**, pp. 1641-1648.

Authors

Yi Wang received his B.Sc Engg and M.Sc.Engg Degrees from Hefei Electronic Engineering Institute in 2004 and 2008, respectively. He is currently a PhD Research student in Hefei Electronic Engineering Institute, Hefei, China. His research interests including speech recognition, machine learning.
E-mail: wygggg@ 126.com

Jun-an Yang received his B.Sc Engg from Southeast University in 1986 and M.Sc.Engg Degrees from Hefei Electronic Engineering Institute in 1991. He obtained his Ph.D (Engg) Degrees from University of Science and Technology of China in 2003. He is currently a Professor in Hefei Electronic Engineering Institute, Hefei, China. His research area including signal processing, pattern recognition and speech recognition.
E-mail: yangjunan@ ustc.cn

Hui Liu received his B.Sc Engg from Wuhan University in 2004. He obtained his M.Sc.Engg and Ph.D (Engg) Degrees from Hefei Electronic Engineering Institute in 2008 and 2011 respectively. He is currently a researcher in Hefei Electronic Engineering Institute, Hefei, China. His research interests including speech recognition, machine learning and signal processing.
E-mail: christ592604@yahoo.com.cn