The Application of an Improved Deep Belief Network in BLDCM Control System

Jing Li, Yang Tian*, Quan-tie Wang, Yong-jun,Wang

Liaoning Engineering Vocational College, Tieling,Liaoning,112008,china

***Abstract* — Deep belief network (DBN) is a kind of recurrent neural network superposed by several layers of unsupervised restricted Boltzmann machines (RBMs). To solve the defect of RBM during sample training that the fixed learning rate is unfavorable to find the optimal value, a principle of dynamical learning rate was proposed to improve the RBM network so as to increase the accuracy of eigenvector mapping. Moreover, the authors also constructed a DBN with two layers of RBMs, of which the neural networks were self-trained by using contrastive divergence (CD). The improved network was applied to control BLDCM..The experimental results indicate that the improved DBN can effectively accelerate response and also improve the accuracy.**

Keywords - deep belief network (DBN); Deep learning; RBM network; BLDCM

I. INTRODUCTION

Neural network is a working pattern which converts the network learning into a working mode similar to the human brain by simulating the working process of the human brain. According to the anatomical knowledge [1], the researchers for neurosciences learnt that there is an obvious hierarchical structure in the human perceptual system. In other words, in the process of recognizing objects, human brain identifies an object in the light of the processed information by aggregation and decomposition rather than directly projecting the received external signals on the retinas. Thus, the function of visual cortex is to classify the external information and extract features [2]. This kind of working mode effectively increases the working efficiency of human brain, reduces the computational burden and stores useful structural information of the object as much as possible. The theoretical and experimental basis for deep learning are as follows: learner will meet problems due to insufficient depth; while the human brain is a deep structure; moreover, the process of cognition is carried out and abstracted by layer. However, the depth of a learner depends on its structure. If the learner is considered as a directed flow chart, its depth is the longest path from the input node to the output node. For example, the depth of multilayer feed forward neural network is that the number of hidden layers plus 1, namely the output layer. If a depth refers to learning knowledge for one time, the depth of learning is the learning times of the original data by layers. For years, the workers of neural network have been committed to studying the neural networks with deep architecture.

The deep learning model for neural networks can extract different features of the observed objects. For this purpose, deep machine learning models are usually with layered structure, where one or several features from different aspects are extracted from each layer, and the extracted features are served as the input of the next layer. Deep learning has been successfully used for various pattern classifications. For instance, in signal processing, it is utilized to voice [7], image and video, text, language and semantic information which can be obtained by human beings [8-10]. At present, DBN is still in its infancy, so there are some particular tasks which are unsuitable for processing. Besides, there is no effective and parallelizable training algorithm to preferably extract features. Moreover, the convergence rate fails to be improved for the lack of effective and feasible strategies. Thereby, the application of DBN in control system often leads to unsatisfied results.

As for the problems mentioned above, a strategy with dynamical learning rate was put forward to train RBM networks. In the method, the shortcoming of the low precision for extracting features induced by the fixed learning rate was overcome by gradually adjusting the learning rate in the process of training to improve the mapping ability of eigenvector of RBM and DBN. Finally, the learning algorithm was applied to the experiment for controlling BLDCM system. The experimental results manifest that the accuracy has been improved significantly.

II. DEEP BELIEF NETWORK (DBN)

DBN is composed of a series of superimposed RBMs and a layer of BP network (Figure 1). The training process of DBN can be divided into two steps: firstly, the RBM on each layer is trained by using unsupervised learning method and the input of each RBM is the output of the next RBM, i.e., the RBM on each layer has to be trained separately to retain as much feature information as possible while guaranteeing that the eigenvectors are mapped to different feature spaces. Secondly, the BP network of the last layer is utilized to receive the output of the last RBM to train the whole network in a supervised way, meanwhile, the network is tuned slightly. In the first step, the training of the RBM on each layer can ensure to optimize the mapping of the weight in the RBM layer to eigenvector, but it fails to make sure that the whole DBN is the optimal. Owing to the false and unimportant information produced in the mapping process cannot be eliminated completely, but several RBMs can gradually reduce the false and unimportant information generated by the previous layer of network, the process can be regarded as the parameter initialization of the last layer of back propagation (BP). In this way, the shortcomings of BP neural network caused by the randomly initialized parameters can be effectively reduced, i.e., the BP network is prone to produce local minimum or take too much time for training. The BP network after the second step can transmit the information to be adjusted to each RBM and the whole DBN from above to below, which has proved that the effect of the whole network is favorable before BP network [3-4]. Therefore, BP plays a role in fine tuning the whole DBN.

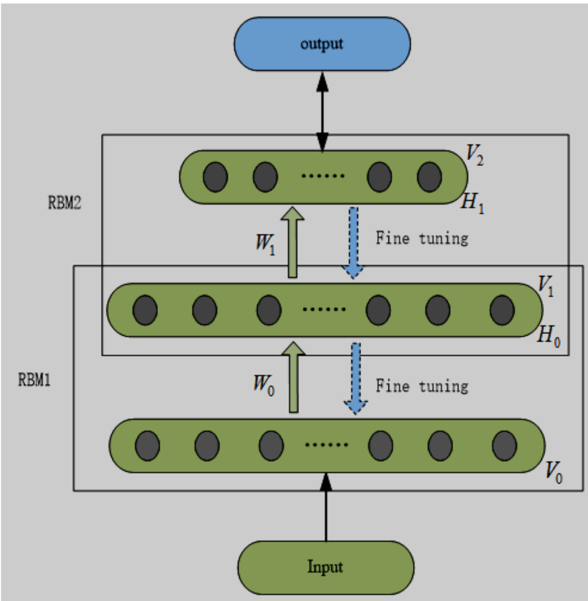

Fig..1 DBN Structure

In the training process of DBN, the bottom network receives the eigenvectors of the original data and transfer them from bottom to top. One or several features of data are extracted at each layer, besides, the extracted features are served as the input of next layer to form combined eigenvectors which are more likely to be classified on the top network. Thus, DBN is a neural network containing many hidden layers and deep architectures. According to the previous research [11], to increase the RBMs in network is still able to ensure the stability of the DBN network. Moreover, more RBMs can make the eigenvector more abstract. Meanwhile, the training results utilized for data classification can be more accurate. In general, the results acquired by using 2~3 RBMs are accurate enough [11].

A. Restricted Boltzmann Machine (RBM) And The Improvement

RBM, as an important part of DBN, is also the core of the whole network. It is made up of a visible layer v and a hidden layer h (see Figure 2), where the former can be considered as the input part and also the output of back propagation. The visible layer and the hidden layer are bi-directionally and fully connected. In addition, the simulated annealing algorithm is adopted as the learning rule. The layers are also connected with each other bi-directionally and fully, but there is no connection in the same layer, which explains why the RBM is "restricted".

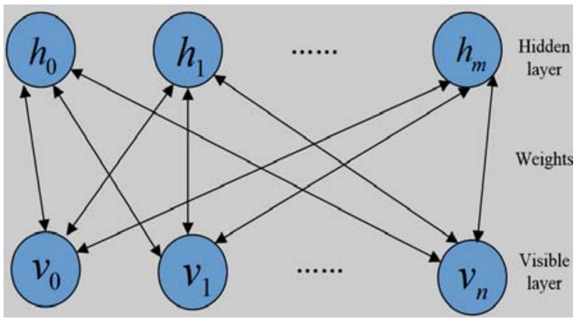

Fig.2 RBM Network Structure

Equation 1 shows the value of the nodes in hidden layer obtained by the known visual layer nodes:

$$p(h_j = 1) = \frac{1}{1 + \exp(-b_j - \sum i v_i \omega_{ij})} \qquad (1)$$

Owing to RBM is a symmetrical network, it can be acquired the value of the visual layer nodes by using the hidden layer nodes, as indicated in Equation 2:

$$p(v_i = 1) = \frac{1}{1 + \exp(-c_i - \sum j h_j \omega_{ij})} \qquad (2)$$

where v_i is the value of the i^{th} node in the visual layer; h_j is the value of the j^{th} node in the hidden layer; b and c are respectively the offset values of the visual layer and hidden layer; and ω_{ij} is the weight between the visual node *i* and the hidden node *j*.

Equation 3 reflects the joint probability distribution between the eigenvector of the visual layer V and the eigenvector of the hidden layer h :

$$p(v,h) \propto \exp(-E(v,h)) = e^{h^T \omega v + b^T + c^T h} \qquad (3)$$

where ω is the weight between the visual layer and hidden layer; $E(v,h)$ is the mathematical expectation of eigenvectors v and h, and its absolute value represents the information of v preserved by h. The parameters needed to be determined in the equation include ω, *b* and *c*, *i.e.*, it is necessary to solve $\theta = (\omega, b, c)$ to maximize the

DOI 10.5013/IJSSST.a.17.31.08 ISSN: 1473-804x online, 1473-8031 print

joint probability distribution in Equation 3, namely $P(v,h)$ [12]. Owing to the Θ satisfying conditions cannot be determined by using the maximum likelihood method, Markov chain Monte Carlo (MCMC) method is employed in traditional strategy to regard the visual layer and hidden layer of RBM as mutual conditions to constantly obtain the update status. When the visual layer and hidden layer reach a steady state together, $P(v,h)$ is up to the maximum. On this basis, the slope of the joint probability distribution between $P(v,h)$ and the initial state, *i.e.*, $\frac{\partial \log p(v,h)}{\partial \theta}$. And then the weight is updated by using Equation 4.

$$\theta^{(\tau+1)} = \theta^{\tau} + \frac{\partial \log p(v,h)}{\partial \theta} \tag{4}$$

where τ is the number of iterations and η is the learning rate. τ can be set according to the network size, say, $\tau = 100 \to 200$; while the learning rate η can influence the learning progress of network, and the appropriate learning rate has to guarantee that $\Theta = (w,b,c)$ which is considered as the necessary condition for the optimal state of learning. In the training process of RBM, the network is adjusted within a wide range at the beginning of operation, while at the late iteration, the network basically tends to be stable and it is only needed to be tuned partially and slightly. It explains that the learning rate η with traditional fixed length is not suitable for iterative optimization. Hence, the learning rate in this experiment was dynamically regulated by setting a learning coefficient μ, and it was adjusted according to the following equation:

$$\eta_t = \mu \eta_{\max} + (1-\mu)\eta_{\min} \tag{5}$$

where μ is the learning coefficient, which is used to control the values of learning rate; $\eta_{\max}$ and $\eta_{\min}$ are the preset upper boundary and lower boundary of the learning rate. Owing to η is generally the reciprocal of τ , $\eta_{\max}$ and $\eta_{\min}$ can be set as $1/\tau$ and $1/2\tau$ respectively.

Thereby, the training process of network can be calculated by using Equation 6:

$$\begin{aligned}\frac{\partial \log p(v,h)}{\partial \theta_{ij}} &= \langle h_j^0 (v_i^0 - v_i^1) \rangle + \langle v_i^1 (h_j^0 - h_j^1) \rangle + \ldots \\ &= \langle h_j^0 v_i^0 \rangle - \langle h_j^0 v_i^1 \rangle + \langle h_j^0 v_i^1 \rangle - \langle h_j^1 v_i^1 \rangle + \ldots \\ &= \langle h_j^0 v_i^0 \rangle - \langle h_j^\infty v_i^\infty \rangle \end{aligned} \tag{6}$$

where $\langle h_j^0 v_i^0 \rangle$ is the mean value of the dot product between the input eigenvector and the corresponding eigenvector in hidden layer; $\langle h_j^\infty v_i^\infty \rangle$ is the average value of the product between the eigenvector in the visual layer at the end of Markov chain and the corresponding eigenvector in hidden layer, and it is convergent. Equation 6 explains that the slope of joint probability distribution is independent of intermediate state, but it is merely related to the initial state and final state of the network. Furthermore, the modified parameter Θ can be acquired in accordance with Equation 5 to achieve the purpose of self-training. Thereupon, the updating formula of weight is as follows:

$$\begin{aligned}\theta^{(\tau+1)} &= \theta^{(\tau)} + \eta \frac{\partial \log p(v,h)}{\partial \theta} \\ &= \theta^{(\tau)} + (\eta \max - \eta \frac{(\eta \max - \eta \min)}{\tau \max}) \\ &(\langle h_j^0 v_i^0 \rangle - \langle h_j^\infty v_i^\infty \rangle)\end{aligned} \tag{7}$$

In fact, the step length is difficult to determine, so the convergence of joint probability distribution $p(v_i^\infty, h_j^\infty)$ between $p\ (v\ ,\ h)$ obtained by using Markov chain and the end is hard to be guaranteed. Thus, CD principle [13] could be adopted in the experiment to improve the computing speed and guarantee the computing accuracy. The equation for the CD principle is as follows:

$$CD_n = KL(P_0 \| P_\infty) - KL(P_n \| P_\infty) \tag{8}$$

where P_0 is the joint probability distribution of RBM when t=0, namely the initial state; P_n is the joint probability distribution of the network after the calculation of Markov chain for *n* step; P_∞. is the distribution at the end of Markov chain; if $KL(P \| P')$ stands for the differences between P and P', CD_n is utilized by P_n to measure the position between P_0 and P_n. In the operation, P_n. is continuously assigned to P_0 to acquire new P_o and P_n .

In the experiment, letting $n = 1$, it could be obtained that the value of CD_n tended to 0 after solving the slope for γ times and updating Θ by using Equation 4. Moreover, the accuracy of the experimental results was approximate to the results obtained by using MCMC method.

As for the output layer of RBM, if s_j stands for the neuron j , in the hidden layer, Equations 9 and 10 respectively represent the learning process of the improved RBM.

$$s_j = \varphi_j \left(\sum i \omega_{ij} s_i + \sigma N_J(0,1)\right) \tag{9}$$

Meanwhile,

$$\varphi_j(x_j) = \theta_L + (\theta_H - \theta_L) \bullet \frac{1}{1 + e^{-a_j x_j}} \tag{10}$$

DOI 10.5013/IJSSST.a.17.31.08

ISSN: 1473-804x online, 1473-8031 print

where $N(0,1)$ stands for the Gaussian random variable with an average value of 0 and the variance of 1; σ is a constant; $\varphi()$ is the Sigmoid function with asymptotic lines of and Θ_H and Θ_L; a is a variable controlled by noise, which means the control on the slope of Sigmoid function. Based on the CD principle, the updating formula for weight and offset value is:

$$\Delta\omega_{ij} = \eta_\omega(\langle s_i^0 s_j^0\rangle - \langle s_i^1 s_j^1\rangle) \tag{11}$$

where η_ω is the learning rate, and $\langle\ \rangle$ stands for the same definition mentioned above.

B. The Adjustment Process Of BP Network

BP network plays a role of supervision and classification in DBN, which receives the output eigenvector of the last RBM and classifies data to slightly adjust the DBN. The training process of BP network mainly contains two steps. Step 1: forward propagation, which refers to spreading eigenvectors to the output end along the input end; Step 2: back propagation, *i.e.*, to obtain the errors by comparing the output results of BP network and the tagged correct results. And then the errors are propagated to the input end from the output end so as to modify the parameters of DBN. In the experiment, Sigmoid function was employed as the function for evaluating the BP network nodes according to the following steps:

(1)To initialize the parameters of BP network to set the step length A;

(2) To calculate in a feed-forward way. As for the j j^{th} element nodes in the l^{th} layer, there is:

$$y_j^l(a) = \sum \omega_{ij}(a) y_i^{l-1}(a) \tag{12}$$

If the neuron belongs to the output layer 1 – L , then letting $y_j^l(a) = O_j^a$, the error is $e_j(a) = d_j(a) - O_j^a$, where d_j is the correct information.

(3) To compute σ, the back propagation is adopted to modify the weight parameters of the network from top to bottom. As for the output elements, there is:

$$\delta_j^l(a) = e_j(a) o_j(a)[1 - o_j(a)] \tag{13}$$

While in terms of the hidden layer elements, there is:

$$\delta_j^l(a) = y_j^l(a)[1 - y_j^l(a)] \sum \delta_k^{i+1}(a)\omega_{kj}^{i+1}(a)$$

(4)To modify the weight.

$$\omega_{ji}^l(a+1) = \omega_{ji}^l(a) + \eta\delta_j^l y_i^{l-1}(a) \tag{14}$$

where η is the learning rate;

(5) (5) If $a = A$, the training is terminated; otherwise, $a = a + 1$, return to Step 2.

III. EXPERIMENT AND ANALYSIS

A. Experimental Settings

Galil controls the brushless direct-current motor control system (BDMCS) in a numerically-controlled machine tool [55]. The control process can be embodied as illustrated in Figure (15), where the controller consists of the improved DBN neural network, digital analog converter (DAC) and zero-order holder (ZOH). In addition, AMP is an error amplifier acting on the brushless direct current motor, which is produced by PWM drive circuit. If the relevant parameters are preset, the closed-loop transfer function for the brushless motor control system can be obtained. And then PID is applied to adjust signals to make each closed-loop system acquire an ideal input function so that the output of the whole motor control system can achieve a desired state. Moreover, DAC is employed to transform the digital quantity into analog quantity by using a 16-bit register; while the cycle of ZOH depends on the sampling period of controller, namely *T*. The corresponding functions and variables for each component are as follows:

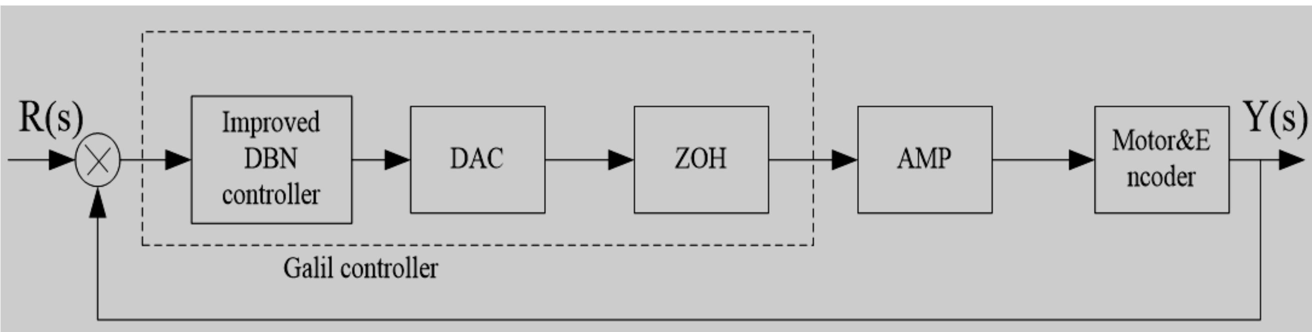

Fig..3 PID Control System

The transfer function for ZOH is:

$$H(s)=\frac{1}{1+\frac{TS}{2}}(T=0.001ms) \quad (15)$$

The range of the analog voltage for DAC input is ± 10, then:

$$k_d=\frac{20v}{2^N cnt}=\frac{20v}{2^{16} cnt}=0.003(v/cnt) \quad (16)$$

AMP:

$$k_a=1.2amp/v \quad (17)$$

Motor & Encoder is composed of encoder and motor, where the parameter for the encoder is:

$$k_f=\frac{4N}{2\pi}=\frac{4\times 1000}{2\pi}=636 \quad (18)$$

Based on Equations (15)and (16), the mathematical model for the brushless direct current motor is:

$$\frac{\theta(s)}{v(s)}=\frac{\frac{k_j}{R_a+J}+K_f}{s+\left[s+\frac{1}{J}(B+\frac{K_f^2}{R})\right]}=\frac{609.8}{s^2+126.8s}$$

where $k_f=\frac{4N}{2\pi}(N=1000)$ is the parameter of encoder; the electrical resistance, R_a is 2.58 Ω; the rotor inertia, J of 1.30E-04; the torque coefficient K_T of 0.208; and viscous damping B of 3.78E-03.

When the parameters were set, the transfer functions for each link in Figure (3) are listed according to relationships between each other.

Through running the program, the closed-loop transfer function for the system can be obtained as follows:

$$T_s=\frac{4.154e005s^2+4.524e007s+1.076e009}{s^4+2.127s^3+6.69e005s^2+4.524e007s+1.076e009} \quad (19)$$

On this basis, the corresponding step response curve is shown in Figure (4).

Figure(4) indicates that the overshoot of the system is:

$$\delta\% = \frac{1.09-1}{1} \times 100\% = 9\%$$

The damping ratio ξ can be calculated by using the following equation:

$$\delta\% = 100e^{-\xi\pi / \sqrt{1-\xi^2}} \Rightarrow \xi = 0.612$$

So:

$$T_p = \frac{\pi}{\omega_n \sqrt{1-\xi^2}} (T_p = 0.346) \Rightarrow \omega_{np} = 105$$

$$T_S = \frac{4}{\xi\omega_n} (T_S = 0.0925) \Rightarrow \omega_{nS} = 70.18$$

The experiment was conducted as the following steps: first of all, a DBN with two layers of RBM was established, where each RBM contained 100 neurons. And then 2000 input including R(s) and Y(s) and output data (Y(s+1)) were utilized to train RBMJ for 50 iterations, where the learning rate was 0.01 ~ 0.02. With the same iterations and learning rate, the network was tuned slightly by using BP algorithm. Finally, the trained DBN was employed to control and forecast 1000 experimental groups, where R(s) and Y(s) are included.

The experimental results are illustrated in the following figure.

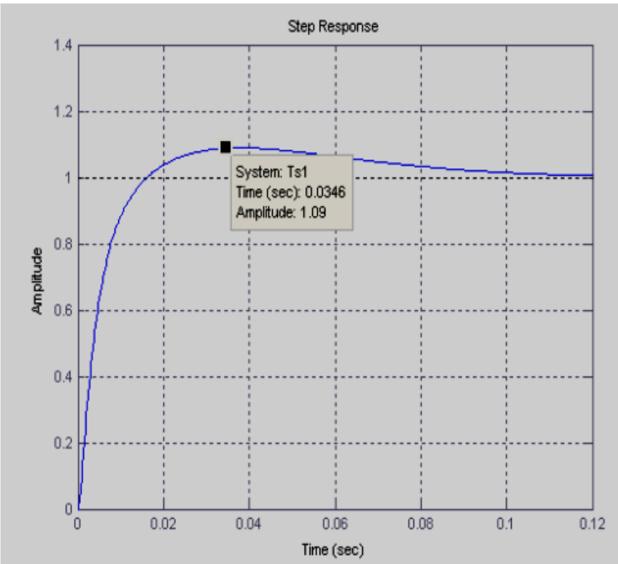

Fig.4 The Step Response Of The BDMCS By The Regulation Of PID Controller

The black bold part in the table shows the results by using the improved network.

TABLE I. THE EXPERIMENTAL RESULTS

Control method	Hidden layer neuron	Response time	Error
PID	*	0.0825	0.707
DBN	100*2	0.0653	0.672
Improved DBN	**100*2**	0.0346	0.582

The two experiments above demonstrate that the response time of the neural network in the improved way is shorter than that of the original network. In addition, the training precision of DBN is improved greatly by superimposing two layers of RBM. Thus, the validity of the improved neural network is validated.

IV. CONCLUSIONS

The rise and development of DBN were discussed first, and then the authors introduced the network structures and algorithms, and also analyzed the superiority of the algorithms. With regard to the low accuracy in seeking optimal values induced by the fixed learning rate of the algorithm, the algorithm was impoved. In addition, the function for output layers was modified to apply the network to the control experiment for electric machine tool system. The experiment results indicate that the improved strategy can effectively reduce the response time of system and improve the accuracy. Furthermore, the modified DBN can be used to the tasks, such as image processing, pattern classification, automatic control, which is regarded as a kind of applicable algorithm in control field.

ACKNOWLEDGMENT

This work is supported by the general project of Education Department of Liaoning Province (No.L2012497) and the general project of Education Department of Liaoning Province (No.L2014589)

REFERENCES

[1] LEETS, MUMFORD D. Hierarchical Bayesian inference in the visual cortex. Journal of optical society of America,20(7): 1434-1448,2003.

[2] ROSSI A F, DESIMONE R, UNGERLEIDER L G. Contextual modulation in primary visual cortex of macaques. Journal of neuroscience,21(5): 1689-1709,2001.

[3] Hinton G E, Osindero S and the Y. A fast learning algorithm for deep belief nets. Neural computation,18: 1527-1554,2006.

[4] Yoshua Bengio, Pascal Lamblin, Dan Popovici and Hugo Larochelle. Greedy layer-wise training of deep networks. Advances in neural information processing systems 19 (NIPS 2006), MIT Press, 153-160,2007.

[5] Marc' Aurelio Ranzato, Christopher Poultney, Sumit Chopra and Yann LeCun. Efficient learning of sparse representations with an energy-based model. Advances in neural information processing systems (NIPS 2006),MIT Press,2007.

[6] Hinton G E and Salakhutdinov R R. Reducing the dimensionality of data with neural networks. Science,313(5786): 504–507,2006.

[7] DAHL G E, YU Dong, DENG Li, et al. Large vocabulary continuous speech recognition with context-dependent DBN-HMMS. Proc of IEEE international conference on acoustics, speech and signal processing. 4688-4691,2011.

[8] DESELAERS T, HASAN S, BENDER O, et al. A deep learning approach to machine transliteration. Proc of the 4th workshop on statistical machine translation.233-241,2009.

[9] FASEL I, BERRY J. Deep belief networks for real-time extraction of tongue contours from ultrasound during speech. Proc of the 20th International Conference on Pattern Recognition. Stroudsburg, PA: Association for computational linguistics,1493-1496,2010.

[10] DENG Li, SELTZER M L, YU Dong, et al. Binary coding of speech spectrograms using a deep auto-encoder. Proc of the 11th annual conference of international speech communication association. 1692-1695,2010.

[11] Y Bengio. Learning deep architectures for AI. Foundations & trends in machine learning, 2009.

[12] Hinton GE. Products of experts. Proc. of the 9th Int'l conf on artificial neural networks (ICANN), Edinburgh, Vol.1:1-6,1999.

[13] [13] Hinton GE. Training products of experts by minimizing contrastive divergence. Neural computation,14(8):1771- 1800,2002.