Research of the DBN Algorithm Based on Multi-innovation Theory and Application of Social Computing

Pinle Qin[1(✉)], Meng Li[1], Qiguang Miao[2], and Chuanpeng Li[1]

[1] School of Computer and Control Engineering,
North University of China, Taiyuan 030051, China
QPL@nuc.edu.cn
[2] School of Computer Science and Technology,
Xidian University, Xian 710075, China

Abstract. Aimed at the problems of small gradient, low learning rate, slow convergence error when the DBN using back-propagation process to fix the network connection weight and bias, proposing a new algorithm that combines with multi-innovation theory to improve standard DBN algorithm, that is the multi-innovation DBN(MI-DBN). It sets up a new model of back-propagation process in DBN algorithm, making the use of single innovation in previous algorithm extend to the use of innovation of the preceding multiple period, thus increasing convergence rate of error largely. To study the application of the algorithm in the social computing, and recognize the meaningful information about the handwritten numbers in social networking images. This paper compares MI-DBN algorithm with other representative classifiers through experiments. The result shows that MI-DBN algorithm, comparing with other representative classifiers, has a faster convergence rate and a smaller error for MNIST dataset recognition. And handwritten numbers on the image also have a precise degree of recognition.

Keywords: DBN algorithm · Convergence error · Multi-innovation theory · MI-DBN algorithm · Social computing

1 Introduction

With the accelerating progress of information digitization and information networking, humans' behavior are recorded more frequently. As a result, it's possible to observe and research the society by computer technology. Computational Social Science is growing up and people will automatically collect and utilize information, the depth and scope of which has never seen before, to serve the social science research. Social computing [1] refers to such a research field combined with computing systems and social behavior, which is investigated how to use computing systems to help people communicate and collaborate, and how to study the rules and trends of social functioning by utilizing computing technology. Social networking services, collective wisdom, social network analysis, content computing and artificial society are included in this kind of research. The issues of content computing consists of public opinion

W. Che et al. (Eds.): ICYCSEE 2016, Part I, CCIS 623, pp. 577–590, 2016.
DOI: 10.1007/978-981-10-2053-7_51

analysis, interpersonal relationships excavation and weibo applications. Microblog simultaneously has the properties of social network and media platform, which triggers the revolution of information production and propagation mode. So images with numbers such as phone numbers and geographic coordinates may include important and useful information. We accurately and quickly recognize the numbers by means of deep learning [2].

Deep learning has achieved great breakthroughs and success in the recent 10 years in many fields such as artificial intelligence, speech recognition, natural language processing, computer vision, Image and visual analysis, multimedia and so on. There are many kinds of models in Deep Learning. One of them, called Deep Belief Networks (DBN) [3, 4], which is an unsupervised algorithm, can create layers of feature detection without requiring labelled data. And the network layers can be used for the reconfiguration and the modeling for the movement of feature detection. During the process of the pre-training, the weights of a deep network can be initialized to meaningful values. After that, a final layer of output units will be added to the top of the network and the whole deep system could be using standard back propagation algorithm. All this operations will be prominently effective in handwritten digit recognition.

Hinton presented DBNs and used it in the task of digit recognition on MNIST dataset [3]. He put forward the DBN network which takes "784-500-500-2000-10" as its structure. Based on the images with 28*28 pixel resolution in MNIST dataset, we can get 784 features in the first layer of the network. The last layer is related to 10 digit labels and other three layers are hidden layers with stochastic binary neurons. Finally this paper achieved 1.25 % classification error rate on MNIST test dataset. Keyvanrad [5] improved the sampling method of Restricted Boltzmann Machines (RBM) based on the DBN proposed by Hinton and get a 1.11 % classification error rate by changing the original Contrastive Divergence (CD) algorithm to Free Energy in Persistent Contrastive Divergence (FEPCD) algorithm. Liu [6] put forward a new categorizer based on DBN, which is the Discriminative Deep Belief Network (DDBN), which integrates the abstract ability of DBN with the resolution capability of back propagation strategy.

Hinton used DBN as a nonlinear model for feature extraction and dimension reduction [7]. Indeed the DBN may be considered as a model that can generate features in its last layer with the ability to reconstruct visible data from generated features. When a general Neural Network is used with many layers, the Neural Network becomes trapped in local minima and the performance will decrease. As a result, it is vital to ensure the initial weights of the nerve net.

According to the multi-innovation identification theory proposed by Ding [8, 9], this paper puts forward a new algorithm–DBN algorithm based on multi-innovation theory. Compared with the previous one, this algorithm accelerates the convergence of errors and increase the accuracy of numerical recognition. The rest of this paper is organized as follows: Sect. 2, Introduce the procedure of original DBN algorithm; Sect. 3, Introduce the multi-innovation identification theory and the improvement of original DBN algorithm; Sect. 4, Introduce the MNIST dataset and the image of the weibo, and test the error date; Sect. 5, Conclude this paper.

2 DBN

DBN is a probability generation model, which is compared with the traditional model of neural network discrimination, and generation model is to establish a joint distribution between the observed data and labels. DBN network structure is composed of several layers of RBM [10] which each layer of output as the input vector to the next layer. The DBN of connection is determined by top-down generation weights, RBM is more easy to connect the weight of learning [11]. DBN stratified training, a tag set is attached to the top. Through a bottom-up, the learned identify the right values obtained a network classifier. Using back propagation technique and using the entire classification equipment adjust the weights optimization classification. Structure is as shown in Fig. 1:

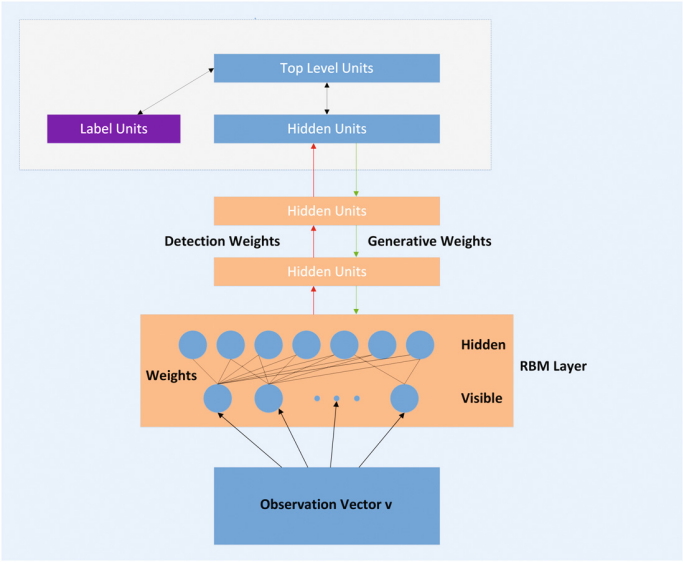

Fig. 1. DBN network structure

The feature extraction of DBN network model requires three process. They are pre-training process, fine-tuning process and the testing process.

2.1 Pre-training Process

Pre-training process is that after extracting input data characteristics through each layer of the network obtain excitation response and add it to the output layer. The specific process is as follows:

(1) First pre-propagation and single unsupervised training of each layer of RBM network. Make sure the feature vector mapped to different characteristics space. Feature information are preserved as much as possible.

(2) The last layer of DBN set a classifier, receiving an output feature vector of RBM as its input feature vector. It supervised to train the entity relationship classifiers. And each layer of RBM network can confirm the weight of its own layer, on the layer of the feature vector to achieve the best, not to the entire DBN feature vector mapping to achieve optimal. Back propagation will transmits the wrong message to every layer of RBM from the top-down, so fine-tune the entire DBN network. The process of RBM network training model can be seen as initialization of weights parameter for logistic regression network. So DBN overcame the shortcomings of logistic regression network that the random initialization parameter weights are easy to fall into the disadvantaged of local optimum and long training time.

2.2 Fine-Tuning Process

Fine-tuning can greatly enhance the performance of a self-encoded neural network. All layers of the network are regarded as a model. In each iteration, the weight and bias of network will be optimized. Specific process is as follows:

(1) Pre-propagation calculations can obtain the activation value $a_i^{(l)}$ of layer L_2, layer L_3 until layer L_{n_l}. Where $a_i^{(l)}$ represents the activation value of the unit i of the layer l (output value). Given a given set of parameters W, b, network can follow function $h_{W,b}(x)$ to calculate output result. $h_{W,b}(x)$ represents the corresponding desired output result. Calculating step of forward propagation is as follows:

$$z^{(l+1)} = W^{(l)} a^{(l)} + b^{(l)} \tag{1}$$

$$h_{W,b}(x) = a^{(l+1)} = f\left(z^{(l+1)}\right) \tag{2}$$

Which uses sigmoid function as the activation function:

$$f(z) = sigmoid(z) = \frac{1}{1+e^{-z}} \tag{3}$$

(2) The error function of each output unit can be calculated
(3) The error function of each layer can be calculated
(4) According to the references [12, 13], we know that the update of parameters W and b can be written as:

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha a_j^{(l)} \delta_i^{(l+1)} \tag{4}$$

$$b_i^{(l)} = b_i^{(l)} - \alpha \delta_i^{(l+1)} \tag{5}$$

2.3 Testing Process

After the pre-training and fine-tuning process, DBN network model has extracted feature layer by layer. At the same time parameters W and b have been optimized. By testing sample of dataset we can test DBN network.

3 DBN Algorithm Based on Multi-innovation Theory

This paper introduces the multi-innovation theory aiming at the situation that standard DBN algorithm mentioned in the paper only uses the current data and loses all the previous epochs data when updating the weight and bias of the parameters. Multi-innovation identification has developed a new field of identification. This method has been widely applied in all kinds of estimations of model parameters because it can improve the fall rate of the gradient.

3.1 Multi-innovation Theory

Common identification algorithm is usually single innovation identification methods which only use single innovation to modify the parameters. [14] As for scalar system: [15]

$$y(t) = \varphi^T(t)\theta + v(t) \tag{6}$$

Where $y(t)$ is the output data of the system, $\varphi^T(t)$ is a vector combined by the output data and input data, θ is the final recognized parameter which is a vector, $v(t)$ is the random noise in system with the zero average, which shows the existence of noises in system.

As for parameter vector θ, using the stochastic gradient identification algorithm has the following formula:

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + L(t)e(t) \tag{7}$$

Where $L(t)$ is a gain vector, $e(t)$ is used to describe the output of the error at time t, $e(t)$ can be showed as:

$$e(t) := y(t) - \varphi^T(t)\widehat{\theta}(t-1) \tag{8}$$

Where $\widehat{\theta}(t)$, modifying the identity parameter at time $t-1$ by multiplying gain vector $L(t)$ by scalar innovation $e(t)$, is the estimation at time t based on the recurrence of $\widehat{\theta}(t-1)$.

Developing it, according to the formulas (21) and (22), extending the scalar innovation $e(t)$ to a form of vector. That is multi-innovation. Simultaneously, the other matrix and vector will change in system. To extend the form of gain vector $L(t)$ to matrix. So we get the multi-innovation identification. It can write as:

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + \Gamma(p,t)E(p,t) \tag{9}$$

$\Gamma(p,t)$ is gain matrix, $E(p,t)$ is innovation vector, p ($p \geq 1$) is the length of the innovation. When the p is equaled with 1, it degenerated as a standard innovation.

Multi-innovation make full use of some of the useful information before, which can have better identification effect. This method increases the computation of the algorithm itself, but it can improve the convergence rate, the precision of the experiment in a small time complexity. The specific recurrence and demonstration of Multi-innovation theory can be found in the reference [16].

3.2 The DBN Algorithm Combined with Multi-innovation—MI-DBN

DBN makes the network reach a great initial point with unsupervised and trained layer by layer [17, 18] before using the back propagation algorithm to optimize the network globally. So it can reach a better local minimum point after training. There probably are thousands of the samples and weights in study system. The top layer is added with the labelled sample to train machine. In order to adjust the weight vector correctly, the gradient vector of each weight is calculated by the back-propagation algorithm, which indicates that the error will increase or decrease if the weight is increased by a very small value. Back propagation algorithm can be used to transmit the gradient through every layer of multilayered network repeatedly, from the output of the top layer to the lowest layer, the gradient vector of weight in every layer can be solved after working out the input derivative in every layer. The weight vector can be adjusted in the opposite direction of the gradient vector. The calculation of the stochastic gradient algorithm is small and the convergence rate is slow. In order to improve the convergence rate of the stochastic gradient identification method, the length of innovation is introduced. The weights updating every time are all based on the weight in last epoch in reverse fine-tuning of DBN. So it improves the convergence rate of the error.

If the weight of the network dynamic changes according to some certain law, the mapping relations between input and output can be changed with time when we identify the time-varying system, so it is possible to identify the time-varying system. Because of the difficulty to get the rules of time variation, the length of innovation p is the whole epoch of the network fine-tuning. Time t is the number of epochs. Define the sequence of positive integer $\{t_0, t_1, t_2, \ldots, t_s\}$, satisfy $0 < t_0 < t_1 < t_2 < \ldots < t_s$, and $1 \leq t_s^* = t_s - t_{s-1}$. In the distance of iteration t_s^* seconds, we probably get the useful data in layer l and node i during every $1/N$ s. The innovation vector $E(p,t)$ is produced to correct the parameter θ using the group of data p. At this time, from $t = t_s - p + 1$ to $t = t_s$, there are p groups of sample data,so the output of node j at time t in layer l which is a visual layer :

$$A_j^{(l)}(p,t_s) = \left[a_j^{(l)}(t_s), a_j^{(l)}(t_s-1), a_j^{(l)}(t_s-2), \ldots, a_j^{(l)}(t_s-p+1)\right]^T \tag{10}$$

The expected output:

$$H_j^{(l)}(p,t_s) = \left[h_j^{(l)}(t_s), h_j^{(l)}(t_s-1), h_j^{(l)}(t_s-2), \ldots, h_j^{(l)}(t_s-p+1)\right]^T \tag{11}$$

The nolinear cost function:

$$J(W,b) = \frac{1}{2}\sum_{j=1}^{l}\left\|H_j^{(l)}(p,t_s) - Y_j^{(l)}(p,t_s)\right\|^2 \tag{12}$$

$Y_j^{(l)}(p,t_s)$ is the P output values of J node in the output layer. $H_j^{(l)}(p,t_s)$ is the corresponding expected output. When calculating output nodes, J will decline, following the rules of network traning, in accordance with gradient in every training cycle. The update of W and b in every iteration are as follows:

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha\frac{\partial J(W,b)}{\partial W_{ij}^{(l)}} \tag{13}$$

Where α is the learning rate. Parameter $W_{ij}^{(l)}$ represents the weights between the unit j of the layer l and the unit i of the layer $l+1$. For each node i of the layer l, we calculate the "residual" $\delta_i^{(l)}$, which shows that the amount of import that the node produces on the error of the final output value. The calculation of $\delta_i^{(l)}$ is based on the error of the weighted average of the layer $l+1$ node. These nodes use $a_i^{(l)}$ as the input value. $\delta_i^{(l)}$ has the following formula:

$$\frac{\partial J(W,b)}{\partial W_{ij}^{(l)}} = \frac{\partial J(W,b)}{\partial z_i^{(l+1)}}\frac{\partial z_i^{(l+1)}}{\partial W_{ij}^{(l)}} \tag{14}$$

It is seen by the formula (1),

$$z_i^{(l+1)} = \sum_{k=1}^{s_l} W_{ik}^{(l)} a_k^{(l)} + b_i^{(l)} \tag{15}$$

So there is that:

$$\frac{\partial z_i^{(l+1)}}{\partial W_{ij}^{(l)}} = a_j^{(l)} \tag{16}$$

Then, the formula (14) is extended to a vector as follows:

$$\frac{\partial J(W,b)}{\partial W_{ij}^{(l)}} = \frac{\partial J(W,b)}{\partial Z_i^{(l+1)}(p,t_s)}A_j^{(l)}(p,t_s) \tag{17}$$

There are:

$$Z_j^{(l)}(p,t_s) = \left[z_j^{(l)}(t_s), z_j^{(l)}(t_s-1), z_j^{(l)}(t_s-2), \ldots, z_j^{(l)}(t_s-p+1)\right] \tag{18}$$

The back propagation of error in the output layer nodes:

$$\delta_i^{(l)} := \frac{\partial J(W,b)}{\partial z_i^{(l)}} \tag{19}$$

The formula (19) is extended to a vector

$$\delta_i^{(l)}(p,t_s) = \frac{\partial J(W,b)}{\partial Z_i^{(l+1)}(p,t_s)} \tag{20}$$

(1) Based on the inference [19], in every the unit of output i error formula can be derived as:

$$\delta_i^{(n_l)} = \frac{\partial J(W,b;x,y)}{\partial z_i^{(n_l)}} = -(y_i - a_i^{(n_l)}) \cdot f'(z_i^{(n_l)}) \tag{21}$$

Formula (3) on both sides at the same time derivative as:

$$f'(z_i^{(n_l)}) = f(z_i^{(n_l)})[1 - f(z_i^{(n_l)})] = y_i \cdot (1 - y_i) \tag{22}$$

Combination (21) and (22)

$$\delta_i^{(n_l)} = \frac{\partial J(W,b)}{\partial z_i^{(n_l)}} = -(y_i - a_i^{(n_l)}) \cdot y_i \cdot (1 - y_i) \tag{23}$$

The formula (23) is extended to a vector

$$\overline{\delta}_i^{(n_l)}(p,t_s) = \frac{\partial J(W,b)}{\partial Z_i^{(n_l+1)}(p,t_s)} = -(Y_j^{(n_l)}(p,t_s) - A_j^{(n_l)}(p,t_s)) \cdot Y_j^{(n_l)} \cdot (1 - Y_j^{(n_l)}) \tag{24}$$

There are

$$\overline{\delta}_i^{(n_l)}(p,t_s) = \left[\delta_i^{(n_l)}(t_s), \delta_i^{(n_l)}(t_s-1), \delta_i^{(n_l)}(t_s-2), \ldots, \delta_i^{(n_l)}(t_s-p+1)\right] \tag{25}$$

Weight of the output layer

$$W_{ij}^{(n_l)}(t_s) = W_{ij}^{(n_l)}(t_s - 1) - \alpha A_j^{(n_l)}(p, t_s)\overline{\delta}_i^{(n_l)}(p, t_s) \tag{26}$$

(2) According to the inference of reference [19], layer l $(l = n_l - 1, n_l - 2, n_l - 3, \ldots, 2)$ can be known. The deviation formulation of the i node in layer l can be derived as:

$$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)}\delta_j^{(l+1)}\right) \cdot f'(z_i^{(l)}) \tag{27}$$

The same as formula (22):

$$f'(z_i^{(l)}) = a_i^{(l)} \cdot (1 - a_i^{(l)}) \tag{28}$$

Referring to the formulas (26) and (27), the deviation function can be extended to the vectors:

$$\overline{\delta}_i^{(l)}(p, t_s) = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)}\delta_j^{(l+1)}\right) \cdot A_i^{(l)}(p, t_s)[1 - A_i^{(l)}(p, t_s)] \tag{29}$$

So the weight of each hidden layer is corrected:

$$W_{ij}^{(l)}(t_s) = W_{ij}^{(l)}(t_s - 1) - \alpha A_j^{(l)}(p, t_s)\overline{\delta}_i^{(l+1)}(p, t_s) \tag{30}$$

(3) Just like the process of correcting and derivating the weight, we can deduce the bias correction function in output layer

$$b_i^{(n_l)}(t_s) = b_i^{(n_l)}(t_s - 1) - \alpha\overline{\delta}_i^{(n_l)}(p, t_s) \tag{31}$$

The bias correction function of each hidden layer:

$$b_i^{(l)}(t_s) = b_i^{(l)}(t_s - 1) - \alpha\overline{\delta}_i^{(l+1)}(p, t_s) \tag{32}$$

Compared with the standard DBN, advantages of the MI-DBN mentioned in this paper are as follows:

(a) When updating the weight and bias, standard DBN just uses the innovation of current epoch while the MI-DBN mentioned in this paper also uses the innovation of several past epochs, which increases the convergence rate of deviation.

(b) Adding multi-innovation to standard DBN can accelerate the convergence of deviation and increase the accuracy of the algorithm. However, adding too much innovation will increase the calculation quantity and decrease the real-timing instead of making the algorithm better. As a result, the amount of innovation should be considered.

4 Experiment and Analysis

In this paper, the improved algorithm is applied to the MNIST handwritten dataset [20]. And the result of the experiment can be achieve by using the toolbox-DeeBNet toolbox [21] in deep learning.

4.1 Experimental Environment

Experiment of software environment: Windows 7 system, VisualStudio2013, Matable2013.Hardware platform: 2.80 GHz Intel (R) CPU E5-2680 V2, 32 GB of memory.

4.2 MNIST Dataset

In our experiments, the used data is based on digital MNIST handwritten dataset. The image pixels have discrete values between 0 and 255 that most of them have the values at the edge of this interval [22]. The image pixel values were normalized between 0 and 1. The dataset was divided to train and test parts including 60,000 and 10,000 images respectively. As shown in Fig. 2:

Fig. 2. MNIST dataset sample

The judging standard involved in this paper are mainly in the following aspects:

(1) Error rate

Error rate of index is mainly used to measure different classification model for MNIST dataset the degree of error classification. The index is an important indicator used to measure the degree of classification error and the direct relationship between the reliability of the algorithm.

Table 1 shows the different algorithms for MNIST dataset classification error rate [4], a bold data is MI-DBN algorithm of MNIST dataset after the classification error rate. It can be seen that the standard DBN algorithm and two DBN improved algorithm in 60000 sample training, 10000 sample testing, 50 epochs training, 150 epochs testing, compared with a much smaller error rate.

Table 1. Classification error rate of MNIST dataset with different classification algorithms

Classification algorithm	Labels	Error rate (%)
SVM	10	1.4
KNN	10	1.6
DBN	10	1.24
FEPCD-DBN	10	1.11
MI-DBN	**10**	**0.54**

(2) Error convergence curve

The simulation data of the experiment error, in the process of experimental training epochs of 50, fine-tuning the epochs of 150. The following chart is the error convergence analysis of different algorithms. As shown in Fig. 3:

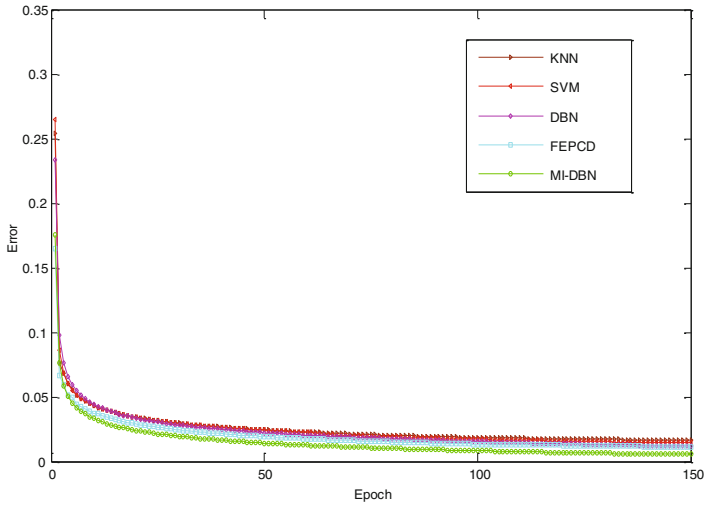

Fig. 3. Error convergence analysis of different algorithms

4.3 Image in Weibo Classification

In recent years, weibo has been developing rapidly. There is a quantity of information in weibo, and it is very hard to efficiently extract useful information we need from massive images in weibo. We created a dataset called WeiBoImage by ourselves. This dataset contains 1000 images downloaded from weibo. Some images of this dataset are as follows Fig. 4. We respectively use trained MI-DBN network and DBN network to extract and recognize handwritten numbers in images.

Fig. 4. Some images of WeiBoImage dataset

First we locate the image and extract information we need. Then we do pixel filling of the image and make the image a 28*28 pixels one after completing the binarization and segmentation of handwritten numbers. Afterwards, the images will be recognized in trained model of MI-DBN or DBN. Specific process is shown as the following Fig. 5:

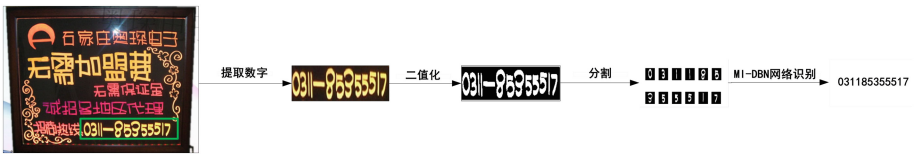

Fig. 5. An image of the specific identification process

The 28*28 image will be the input of the trained network model of MI-DBN and DBN, it is recognized by our trained model. Table 2 shows the MI-DBN algorithm and DBN algorithm of WeiBoImage dataset after the classification error rate.

Table 2. Classification error rate of WeiBoImage dataset with different classification algorithms

Classification algorithm	Error rate (%)
MI-DBN	3.85
DBN	4.5917

It can be concluded from the experiment that when WeiBoImage dataset is tested by network model trained by MI-DBN, the precision of the result is high and the loss of

weibo image information can be solved quickly. This experiment, utilized as a plug-in in social competing software, applies computers to social network service. As a result, the research of this experiment is of great significance.

5 Conclusions

In this paper, based on the deficiency of DBN algorithm, we have presented a new Deep Belief Networks learning algorithm which combines with multi-innovation theory of stochastic gradient identification. Improved algorithm MI-DBN, utilizing historical epochs data, makes the best of useful information implied in the past data, and in this way, the connection weight and offset of the network can reach the predictive value faster. Simulation results show that the error rate of convergence is improved. In the process of MNIST handwritten dataset recognition, the error rate of classification is reduced after applying MI-DBN algorithm. We upload images on weiboa, and identified useful information from the images, we had obtained better effect. It is useful for us, because we will study the images of weibo where reflect the person's emotional state.

Acknowledgment. This work is partially supported by Shanxi Nature Foundation (No.2015011045). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

1. Choi, B.: Multiagent social computing. Int. J. Web Portals **3**(4), 56–68 (2011)
2. Lecun, Y., Bengio, Y., Hinton, G.E., et al.: Deep learning. Nature **521**, 436–444 (2015)
3. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
4. Larochelle, H., Erhan, D., Courville, A., et al.: An empirical evaluation of deep architectures on problems with many factors of variation. In: ICML 2007: 2007 International Conference on Machine Learning, pp. 473–480 (2007)
5. Keyvanrad, M.A., Homayounpour, M.M.: Deep belief network training improvement using elite samples minimizing free energy. Int. J. Pattern Recogn. Artif. Intell. **29**(5), 1411–4046 (2014)
6. Liu, Y., Zhou, S., Chen, Q.: Discriminative deep belief networks for visual data classification. Pattern Recogn. **44**(10–11), 2287–2296 (2011)
7. Hinton, G.E., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2016)
8. Ding, F., Tao, X., Tao, D.: Multi-innovation stochastic gradient identification methods. Control Theory Appl. **20**(6), 870–874 (2003)
9. Ding, F.: Several multi-innovation identification methods. Digital Sig. Process. **20**(4), 1027–1039 (2010)
10. Hinton, G.E.: A practical guide to training restricted boltzmann machines. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade, 2nd edn. LNCS, vol. 7700, pp. 599–619. Springer, Heidelberg (2012)

11. Swersky, K., Chen, B., Marlin, B., et al.: A tutorial on stochastic approximation algorithms for training Restricted Boltzmann Machines and Deep Belief Nets. In: Information Theory and Applications Workshop (ITA), pp. 1–10. IEEE (2010)
12. Schölkopf, B., Platt, J., Hofmann, T.: Greedy layer-wise training of deep networks. Adv. Neural Inform. Process. Syst. **19**, 153–160 (2007)
13. Sarikaya, R., Hinton, G.E., Deoras, A.: Application of deep belief networks for natural language understanding. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(4), 778–784 (2014)
14. Ding, J., Xie, L., Ding, F.: Performance analysis of multi-innovation stochastic gradient identification for non- uniformly sampled systems. Control Decis. **26**(9), 1338–1342 (2011)
15. Ding, F.: Multi-innovation identification theory and methods. J. Nanjing Univ. Inform. Sci. Technol. **4**(1), 1–28 (2012)
16. Ding, F.: Convergence of multi innovation identification under attenuating excitation conditions for deterministic systems. J. Tsinghua Univ. **9**, 111–115 (1998)
17. Lee, H., Ekanadham, C., Ng, A.: Sparse deep belief net model for visual area V2. Adv. Neural Inform. Process. Syst. **20**, 873–880 (2008)
18. Krizhevsky, A.: Learning multiple layers of features from tiny images. University of Toronto, Canada, 17 (2009)
19. Salakhutdinov, R., Hinton, G.E.: Using deep belief nets to learn covariance kernels for gaussian processes. Adv. Neural Inform. Process. Syst. **20**, 1249–1256 (2007)
20. Lecun, Y., Cortes, C.: The MNIST database of handwritten digits [DB/OL] (2011). http://yann.lecun.com/exdb/mnist/index.html
21. Keyvanrad, M.A., Homayounpour, M.M.: A brief survey on deep belief networks and introducing a new object oriented MATLAB toolbox (DeeBNetV2.2). Comput. Vis. Pattern Recogn. **12**, 1408–3264 (2014)
22. Tieleman, T., Hinton, G.: Using fast weights to improve persistent contrastive divergence. In: ICML 2009: Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Quebec, Canada, pp. 1033–1040 (2009)