Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 4174–4179

Improved Classification with Semi-supervised Deep Belief Network

Gongming Wang[1]. **Junfei Qiao**[1]. **Xiaoli Li**[1]. **Lei Wang**[1]. **Xiaolong Qian**[2]

[1]*Faculty of Information Technology, Beijing University of Technology, Beijing, China (e-mail: xiaowangqsd@163.com; junfeiq@bjut.edu.cn; lixiaolibjut@bjut.edu.cn; jade_wanglei@163.com)*

[2]*College of Information Science and Engineering, Northeastern University, Shenyang, China (e-mail: qianxiaolong@ise.neu.edu.cn)*

Abstract: Classification problem is important for big data processing, and deep learning method named deep belief network (DBN) is successfully applied into classification. But traditional DBN is an unsupervised learning method, which leads to a gap between extracted features and concrete tasks. In this paper, a semi-supervised DBN (SSDBN) based on semi-supervised restricted Boltzmann machine (SSRBM) is proposed to shorten the gap and improve the accuracy of classification. Firstly, through introducing relevance constraint, supervised information is equivalently integrated into the learning process of restricted Boltzmann machine. Secondly, SSDBN-based model is constructed to improve the accuracy of classification problem. Finally, the proposed SSDBN is validated with hand-written digits classification standard dataset MNIST, and experimental results show that SSDBN outperforms traditional DBN and other models with respect to classification.

Keywords: Classification problem; deep learning; SSDBN; SSRBM; contrastive experiment

1. INTRODUCTION

Improved classification problem is very important, especially when it comes to the age of big data. In the past few decades, many methods have been applied for classification problems, and the corresponding applications are achieved successfully (Jagtap and Kokare, 2016; Lin et al., 2007). Supervised classification models focus on extracting features only using labelled data, which is usually applied in those classification problems with enough labelled data, such as multilayer perceptron (MLP) (Shekofteh et al., 2015; Gao et al., 2015), support vector machine (SVM)[4] and artificial neural networks (ANN)(Chen et al., 2013; Yuan et al., 2014). But supervised classification models fall into local minimum easily because the initial weights are given randomly (Jagannathan and Lewis, 1996). Additionally, according to the universal approximation theory, if there are sufficient hidden neurons in a single hidden layer, supervised algorithms can model any classification problems to any accuracy (Cybenko, 1989). If the hidden neuron number is equal to the number of training samples, the training error is reduced to zero (Huang et al., 2006). However, the training samples are actually much larger than the hidden neuron number, and most supervised models are shallow architectures. So increasing the number of hidden layers or hidden neurons always leads to exponentially increasing of model complexity, which is unacceptable for classification problem.

Unsupervised classification models, such as fuzzy C-Mean algorithm (FCMA) (Niazmardi et al., 2013) and fuzzy clustering algorithm (FC) (Zhong et al., 2014), only use unlabelled data to extract features. Moreover, unsupervised algorithms always depend on similarity observation, which leads to their sensitivity to noise. From the perspective of reality, the labelled data is always insufficient so that the enough features used to classify are not available, which also makes the unsupervised algorithms hardly achieve acceptable classification performance.

Deep learning is an extension of ANN, its idea of deep learning is to achieve hierarchical representation of original data, then more abstract features and essential attributions are learned. Most importantly, the deep architecture of deep learning networks can extract associated features between input data through the idea of hierarchical representation. Recently, deep belief network (DBN) is one of the most popular deep learning models, and it begins to be used in classification problem in recent years (Shi et al., 2016; Tang et al., 2016; Abdel-Zaher and Eldeib, 2016). DBN is composed of several restricted Boltzmann Machines (RBMs), which are stacked sequentially. RBM is energy-based learning models, the training process of RBM is unsupervised because it only uses input samples. The learning of DBN consists of two stages (Hinton, 2002): (1) Several RBMs stacked sequentially is trained by unsupervised greedy algorithm, this process is called pre-training; (2) After pre-training, error back-propagation based supervised algorithm is applied to adjust the weights generated from pre-training, this process is also called fine-tuning. The pre-training of DBN is equal to random initialization of weights in traditional neural networks, but the initial weights generated by pre-training have already been in a better position. The learning style of DBN is beneficial to overcome local minimum resulted from random initialization. However, although DBN has two learning stages, actually the function of fine-tuning is limited, and DBN is still regarded as an unsupervised learning model (Hinton and Salakhutdinov, 2011), so the absence of supervised data leads to a gap between extracted features and concrete tasks.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.807

With respect to classification problems, the features between input data and supervised data is called associated features. Ideally, a classification model should take the weights initialization from pre-training and the ability to extract associated features into consideration simultaneously. Liu and Zhou have proposed discriminative deep belief networks (DDBN), which uses a new deep architecture to integrate the abstraction ability of deep belief nets (DBN) and discriminative ability of back-propagation strategy (Liu et al., 2011). But DDBN has only focused on the architecture not on improvement for extracting the associated features. (Larochelle and Bengio, 2008) have proposed an evaluation of different learning algorithms for RBMs aiming at introducing a discriminative nonlinear classifiers, although the corresponding improved performance is achieved, actually they have not improved the classification accuracy from whole DBN.

In this paper, inspired by the essence of classification problem, the advantages of supervised and unsupervised algorithm, we combine the unsupervised learning with the supervised learning and propose a new type of classification model named semi-supervised DBN (SSDBN). Particularly, with respect to RBM, in order to simultaneously remain the ability to extract input features and associated features, a relevance constraint is introduced into RBM model to realize that supervised information is used in the process of RBM learning. Because supervised learning and unsupervised learning are both applied to train RBM, this learning method is named semi-supervised pre-training, and the corresponding RBM is named semi-supervised pre-training RBM (SSRBM). SSRBM can be regarded as a new style of RBM, which integrates cognitive model with generative model through hidden layer sharing and weights binding. Likewise, the SSDBN is composed of several SSRBMs stacked sequentially. It should be pointed out that every SSRBM has a supervised information vector, the supervised information vector in different SSDBN layer denotes the corresponding associated feature, which is also the desired associated feature in different layer. Only in this way, the more associated features can be extracted after bottom-up training, and equivalently the output layer of the SSDBN gets all the associated features extracted by all the rest of layers. In the part of experiments, the proposed SSDBN for classification is validated with standard dataset MNIST, and the results show that the SSDBN outperforms traditional DBN and other models with respect to classification.

The rest of this paper is as follows. Section 2 gives the related works. In section 3, the semi-supervised algorithm is presented, including modelling methodology of SSRBM and SSDBN. Section 4 shows the comparison experiments between SSDBN and other methods with respect to classification problem of standard dataset MNIST. The conclusions are given in section 5.

2. RELATED WORKS

2.1 Model of RBM

RBM, a special form of Boltzmann distribution, is composed of visible layer (input) and hidden layer (output). There is a bidirectional full connection between two layers, no connection in same layer, and the RBM structure is shown in Fig.1.

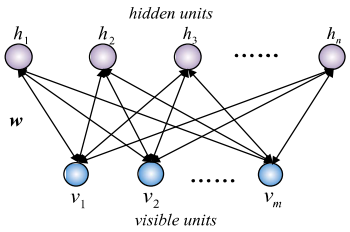


Fig.1 RBM architecture

In Fig.1, let vector $\boldsymbol{v}$ and $\boldsymbol{h}$ represent the state of visible layer and hidden layer respectively, v_i is state of the ith visible node, and h_j is state of the jth hidden node. The outputs of all nodes have only two states 0 and 1, denoting activation and inhibition respectively, then the energy function (Hinton, 2010) based on given configuration is defined as follows:

$$E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta}) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} v_i w_{ij} h_j \tag{1}$$

where $\boldsymbol{\theta} = (w_{ij}, a_i, b_j)$ is RBM's parameter set to be learn, whose elements are real numbers, $\boldsymbol{a}$ and $\boldsymbol{b}$ denote the bias vector of the visible nodes and the hidden nodes respectively, $\boldsymbol{w}$ is a matrix containing the RBM's connection weights. Generally, weights are initialized with small random values, and those bias are initialized with 1.

When those parameters have been determined, the joint probability distribution is given by

$$P(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta}) = \frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})} \tag{2}$$

where Z is partition function, which is described by Eq.(3) from the perspective of physical significance.

$$Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})} \tag{3}$$

The probability distribution of visible $\boldsymbol{v}$, marginal probability distribution of the joint probability distributions, is called likelihood function, which can be given by

$$P(\boldsymbol{v}/\boldsymbol{\theta}) = \sum_{\boldsymbol{h}} P(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h}/\boldsymbol{\theta})} \tag{4}$$

In terms of visible layer and hidden layer in RBM, nodes of same layer are independent. So when state vector $\boldsymbol{v}$ of visible layer has been determined, the activated probability of the jth hidden node is described as

$$P(h_j = 1/\boldsymbol{v},\boldsymbol{\theta}) = \sigma\left(b_j + \sum_{i=1}^{m} v_i w_{ij}\right) \tag{5}$$

Where $\sigma(\cdot)$ is the sigmoid function given by

$$\sigma(x) = \frac{e^x}{e^x + 1} = \frac{1}{1 + e^{-x}} \tag{6}$$

And according to symmetrical structure of RBM, when state vector $\boldsymbol{h}$ of hidden layer has been determined, the

activated probability of the jth visible node is described as

$$P(v_i=1/\boldsymbol{h},\boldsymbol{\theta})=\sigma\left(a_i+\sum_{j=1}^{n}w_{ij}h_j\right) \tag{7}$$

Judging criterion for activation or inhibition is based on the result of comparison between actual probability and presupposed threshold ε. Specifically, h_j is set to 1 when $P(h_j=1/\boldsymbol{v},\boldsymbol{\theta})$ is greater than ε and 0 otherwise, the principle is same when it comes to visible layer, which can be implemented by

$$h_j=\begin{cases}1 & if \quad P(h_j=1/\boldsymbol{v},\boldsymbol{\theta})>\varepsilon\\ 0 & if \quad P(h_j=1/\boldsymbol{v},\boldsymbol{\theta})<\varepsilon\end{cases} \tag{8}$$

Generally, is a random constant uniformly distributed between 0.5 to 1.

2.2 Learning algorithm of RBM

The purpose of training RBM is to find out the best parameters set, so that likelihood function is maximized. In order to solve this problem, Hinton proposed a much faster algorithm of training RBM: the contrastive divergence (CD) algorithm (Hinton et al., 2006). According to CD algorithm, the best parameters set can be obtained by maximizing log-likelihood function, which is based on RBM training set (assuming contains D training samples). For implementation, it is given by

$$\boldsymbol{\theta}^*=\arg\max_{\boldsymbol{\theta}}F(\boldsymbol{\theta})=\arg\max_{\boldsymbol{\theta}}\sum_{d=1}^{D}\log P(\boldsymbol{v}^{(d)}/\boldsymbol{\theta}) \tag{9}$$

Specifically, stochastic gradient ascent is used to solve the maximum. During the process of CD algorithm, the most critical step is to solve partial derivative of $\log P(\boldsymbol{v}^{(d)}/\boldsymbol{\theta})$ with respect to parameters $\boldsymbol{\theta}$.

Hinton has proposed Markov Chain Monte Carlo (MCMC), which can solve the state of visible layer and hidden layer. When the distribution of visible layer and hidden layer comes to steady state, $P(\boldsymbol{v},\boldsymbol{h})$ comes to maximum, the parameters can be updated as follows(Le Roux and Bengio, 2008).

$$\boldsymbol{\theta}^{(\tau+1)}=\boldsymbol{\theta}^{(\tau)}+\eta\frac{\partial\log P(\boldsymbol{v},\boldsymbol{h})}{\partial\boldsymbol{\theta}} \tag{10}$$

where τ and η are iteration number and learning rate respectively.

3. THE PROPOSED METHOD

3.1 Semi-supervised RBM

Considering a set of data $\boldsymbol{V}=\{\boldsymbol{v}_1,\boldsymbol{v}_2\cdots\boldsymbol{v}_N\}$ and a real number matrix $\boldsymbol{S}\in\mathbf{R}^{N\times N}$ is defined to represent associated features.

$$S_{ij}=\begin{cases}\dfrac{\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{v}_j}{\|\boldsymbol{v}_i\|\|\boldsymbol{v}_j\|} & \text{supervised data for } \boldsymbol{v}_i \text{ and } \boldsymbol{v}_j \text{ are different}\\ 0 & \text{supervised data for } \boldsymbol{v}_i \text{ and } \boldsymbol{v}_j \text{ are same}\end{cases} \tag{11}$$

where $i,j=1\cdots N$. When $\boldsymbol{S}$ is given, a set of feature vector $\boldsymbol{H}=(\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_N)$ is defined, and the relevance constraint problem based on the feature vector can be described as

$$\arg\min_{\boldsymbol{H}}\left\|\boldsymbol{S}-\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\right\| \tag{12}$$

Due to the fact that Eq. (12) is a quadratic optimal decision problem with respect to matrix, it is difficult to obtain the solvable energy function based on thermodynamic significance. So the method of singular value decomposition (SVD) is used to transform Eq.(12) into a simple one, and corresponding energy function is available. Assuming that $\boldsymbol{S}=\boldsymbol{G}\boldsymbol{\Sigma}\boldsymbol{G}^{\mathrm{T}}$ is derived from SVD, a definite solution can be described as

$$\boldsymbol{u}=\boldsymbol{A}_m^{1/2}\boldsymbol{G}_m^{\mathrm{T}} \tag{13}$$

where $m<N$, $\boldsymbol{A}_m$ is a diagonal matrix including m large singular values of $\boldsymbol{S}$, $\mathbf{G}_m$ is the left singular value vector. It is not difficult to verify that $\boldsymbol{u}$ is multiplied by any orthogonal matrix $\boldsymbol{P}$ to obtain a matrix, and this matrix is still the solution of Eq. (12), so Eq. (12) is rewritten by

$$\begin{cases}\arg\min_{\boldsymbol{H},\boldsymbol{P}}\left\|\boldsymbol{u}\boldsymbol{P}^{\mathrm{T}}-\boldsymbol{H}\right\|\\ s.t. \quad \boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}=\boldsymbol{E}\end{cases} \tag{14}$$

where $\boldsymbol{E}$ is an identity matrix. For convenience, the constraint condition for matrix $\boldsymbol{P}$ is ignored and Eq. (14) can be simplified as

$$\arg\min_{\boldsymbol{H},\boldsymbol{P}}\left\|\boldsymbol{u}\boldsymbol{P}^{\mathrm{T}}-\boldsymbol{H}\right\| \tag{15}$$

If $\boldsymbol{u}=(u_1,u_2,\cdots,u_m)$ and $\boldsymbol{H}$ are regarded as desired output and actual output respectively, Eq.(15) is considered as a supervised learning problem with single-layered network. According to related conclusions thermodynamics and energy function of RBM, the definition of energy function with respect to $\boldsymbol{u}$ and $\boldsymbol{H}$ is given by

$$E(\boldsymbol{u},\boldsymbol{h}/\boldsymbol{\varphi})=-\sum_{k=1}^{m}c_ku_k-\sum_{j=1}^{n}b_jh_j-\sum_{k=1}^{m}\sum_{j=1}^{n}u_kp_{kj}h_j \tag{16}$$

where $\boldsymbol{\varphi}=(p_{kj},c_k,b_j)$, $\boldsymbol{C}$ denotes the bias of the $\boldsymbol{u}$.

According to Eq. (16), a new model constructed by $\boldsymbol{u}$ and $\boldsymbol{h}$ shares the same hidden layer with traditional RBM, which forms a semi-supervised RBM (SSRBM). The architecture of SSRBM is shown in Fig.2.

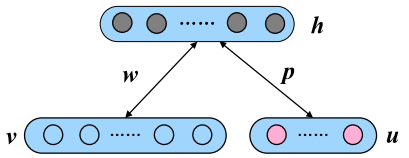


Fig.2. Architecture of SSRBM

The energy function of SSRBM is defined as

$$\begin{aligned}E(\boldsymbol{v},\boldsymbol{u},\boldsymbol{h}/\boldsymbol{\psi})=&-\sum_{i=1}^{m}a_iv_i-\sum_{j=1}^{n}b_jh_j-\sum_{i=1}^{m}\sum_{j=1}^{n}a_iw_{ij}h_j\\&-\lambda\sum_{k=1}^{m}c_ku_k-\lambda\sum_{k=1}^{m}\sum_{j=1}^{n}u_kp_{kj}h_j\end{aligned} \tag{17}$$

where $\boldsymbol{\psi}=(w_{ij},p_{kj},a_i,c_k,b_j)$, λ is weight parameter controlling the proportion of supervised and unsupervised learning. The conditional probability distributions of SSRBM are given by

$$P(h_j/\boldsymbol{v},\boldsymbol{u},\boldsymbol{\psi})=\sigma\left(\sum_{i=1}^{m}v_i w_{ij}+\lambda\sum_{k}^{m}u_k+b_j\right) \tag{18}$$

$$P(v_i/\boldsymbol{h},\boldsymbol{\psi})=\sigma\left(\sum_{j=1}^{n}w_{ij}h_j+a_i\right) \tag{19}$$

$$P(u_k/\boldsymbol{h},\boldsymbol{\psi})=\sigma\left(\lambda\sum_{j=1}^{n}p_{kj}h_j+\lambda c_k\right) \tag{20}$$

So the parameters $\boldsymbol{\psi}=(w_{ij},p_{kj},a_i,c_k,b_j)$ can be updated as follows

$$\boldsymbol{\psi}^{(\tau+1)}=\boldsymbol{\psi}^{(\tau)}+\eta\frac{\partial\log P(\boldsymbol{v},\boldsymbol{u},\boldsymbol{h})}{\partial\boldsymbol{\psi}} \tag{21}$$

where τ and η are iteration number and learning rate respectively.

3.2 Semi-supervised DBN

Like the DBN, SSDBN is also a generative model with latent variables. The architecture of TDBN is shown in Fig.3.

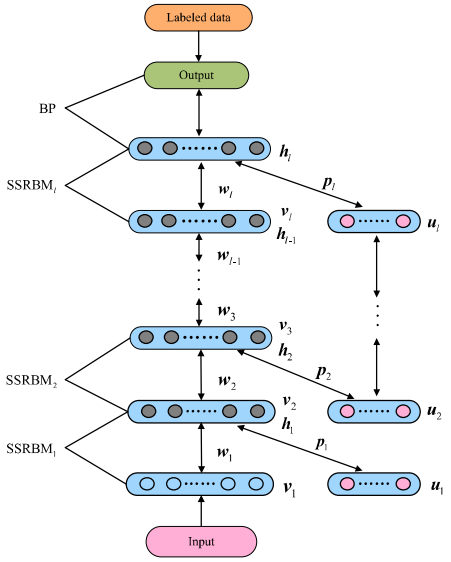


Fig.3. Architecture of SSDBN

The latent variables are typically binary, while the visible units may be binary or real. There are no intra-layer connections, usually, every unit in each layer is connected to every unit in each neighbouring layer, though it is possible to construct more sparsely connected DBNs. The SSDBN is composed of several SSRBMs stacked sequentially, the visible layer of the first SSRBM is regarded as the input layer of SSDBN, the other layers are named as hidden layers. The original data is delivered to SSDBN through the input layer and transformed into an abstract representation by the hidden layers, which is called as a process of encoding. According to concrete tasks, the abstract representation is mapped from the last hidden layer to the output layer, which is called as a process of decoding.

The learning process of SSDBN consists of two parts: semi-supervised pre-training and supervised fine-tuning. Specific pre-training process are listed as follows: (1) The visible layer and supervised layer of the lower-level SSRBM receives samples data and supervised data respectively to start training; (2) The output of lower-level SSRBM acts as the input of the higher-level SSRBM; (3) Repeating the process (1) and (2) to train desired number of layers. Since SSRBM can be fast trained through CD algorithm, greedy layer-wise training avoids the high complexity of training the whole TDBN directly. Actually, semi-supervised pre-training not only is regarded as a process of parameters initialization, but also it has an outstanding ability of extracting associated features. After semi-supervised pre-training, parameters of SSDBN are determined into a small and better range, then traditional global algorithm (BP) is applied to adjust these parameters, which is also called fine-tuning.

4. EXPERIMENT STUDIES

The main purpose of this section is to evaluate and test the performance of the proposed SSDBN with respect to classification. We present an experiments: hand-written digital recognition. In these two experiments, the architecture of SSDBN model is selected by a validation set, which is conducted before testing. Meanwhile, a criterion named as reconstruction error is used to evaluate the performance of semi-supervised learning in SSRBM. The definition of reconstruction error in SSRBM is described as

$$R\text{-}error=\frac{\sum_{i=1}^{s}\sum_{j=1}^{d}\left|v'_{ij}-u_{ij}\right|}{s\cdot d} \tag{22}$$

where s and d are the number of samples and samples dimension, v'_{ij} and u_{ij} are reconstruction data and target extracted feature respectively.

The MNIST database includes 60000 training images and 10000 test images all of which are images of handwritten numbers. The images represent the numerals (from 0 to 9) with size of 28×28 pixels. In this experiment, 5000 randomly chosen samples are used for the training process, and an additional 1000 for the testing process. The SSDBN has two hidden layers, and the number of neurons in hidden layers is 200, the number of semi-supervised training iterations is 100. Fig.4 and Fig.5 are the reconstruction error of bottom SSRBM and top SSRBM respectively, the reconstruction errors decrease very fast in the first 10 iterations, and they converge to 9.3e-4 and 6.8e-5 respectively. Fig.6 (a) is the original images of written digitals, Fig.6 (b) is the classification mistakes for the original images, which shows that the number of classification mistakes is only 48.

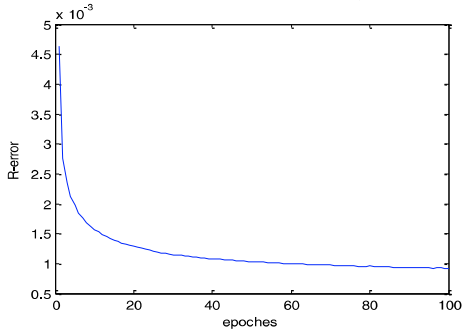


Fig.4. Reconstruction error of bottom SSRBM

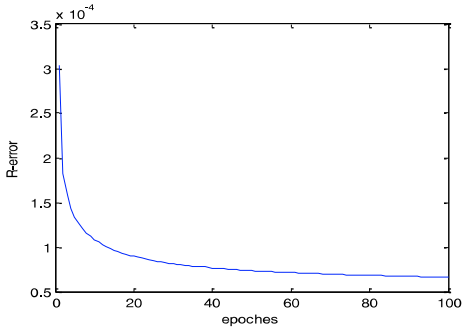
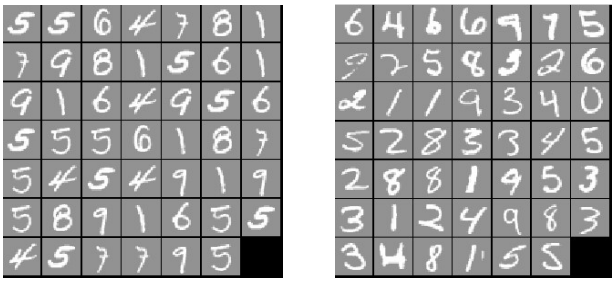


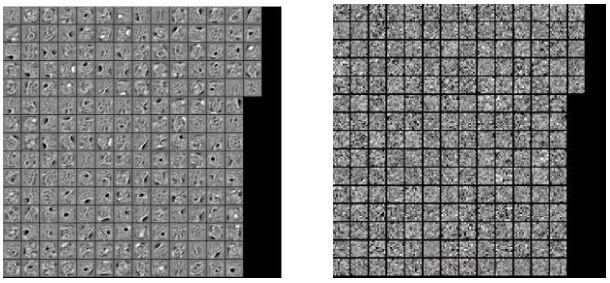
Fig.5 Reconstruction error of top SSRBM



(a) (b)

Fig.6 (a) Original images; (b) Images with classification mistakes

Fig.7 (a) and Fig.7 (b) denote the learned weights of two hidden layers respectively, which are substantially the images of features extracted. According to Fig.7 (a), we can see that the hidden layer1 mainly extracts edge features of original images, so the features image is a dark smooth curve with a vague outline and weakened background. Compared with Fig.7 (a), Fig.7 (b) gives a features image with a more disordered and weakened background, which indicates that the hidden layer2 extracts more abstract features.



(a) (b)

Fig.7 (a) Learned weights in hidden layer1; (b) Learned weights in hidden layer2

Furthermore, Fig.8 gives the number of classification mistakes using different models. Without loss of generality, 50 trials have been performed independently for each model. It can be seen from Fig.8 that, although the classification mistakes of all models have fluctuations in 50 trials, these fluctuations are all within a small range. The main reason for the existence of fluctuations is that the initial weight of pre-training is derived from random initialization. Although the purpose of pre-training is to present an initial weight for fine-tuning, actually random initialization is unavoidable in the beginning of pre-training. In particular, random initialization is unavoidable, not only to the SSDBN, but almost all the neural networks.

In order to specifically demonstrate the superiority of SSDBN regarding classification, Table 1 gives the more details about comparison results among different models, including model architecture, the average number of classification mistakes, average reconstruction error and variance. Particularly, it need to be pointed out that CDBN is continuous DBN, CNN is convolutional neural network, and SVM is support vector machine.

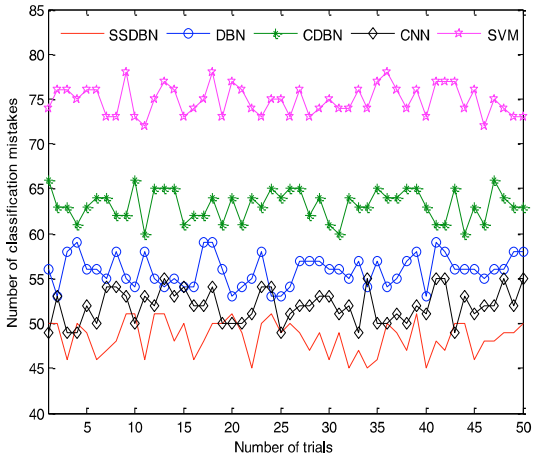


Fig.8. Classification mistakes of different methods

From Fig.8 and Table 1, it is concluded that the proposed SSDBN has a smaller average number of classification mistakes. Meanwhile, it is observed that the reconstruction error and the corresponding variance in SSDBN is smaller than that of any other methods listed in Table 1, which indicates that the performance of SSDBN is relatively better and stable.

Table 1 Performance comparison of classification

Methods	Neurons number	Classification mistakes number*	R-error (bottom)		R-error (top)	
			Average	Variance	Average	Variance
SSDBN	784-200-200-10	**48**	**9.3e-4**	**0.0020**	**6.8e-5**	**0.0019**
DBN	784-200-200-10	56	1.2e-3	0.0037	8.1e-5	0.0039
CDBN	784-200-200-10	63	1.9e-3	0.0041	9.6e-5	0.0040
CNN	784-200-200-10	52	—	—	7.3e-5	0.0022
SVM	—	75	—	—	—	—

Bold values signifies the best results, "*" denotes average value.

5. CONCLUSIONS

In recent years, the problem of classification has been attracting the attention of scholars, especially those researchers specializing in big data analysis. In this paper, a new type of classification model called SSDBN is presented and achieves a better result. The SSDBN based classification model has the following advantages compared with other models.

(1) Deep learning based DBN has two advantages: 1) the weights are guided to regions of minimal norm; 2) it sets the weights in zones of the parameter space, in which the likelihood of a global minimum is maximum; 3) it can sufficiently extract the associated features of all the input data. SSDBN just takes both advantages of the deep learning and SPRBM. As a result, a more accurate result for classification is obtained from SSDBN model.

(2) In the paradigm of semi-supervised learning, $\boldsymbol{u}=(u_1,u_2,\cdots,u_m)$ is regarded as supervised data or labelled data, both unlabelled examples from $P(\boldsymbol{v})$ and labelled examples from $P(\boldsymbol{v},\boldsymbol{u})$ are used to obtain the other representation of $\boldsymbol{v}$. Additionally, in the context of scarcity of labelled data and abundance of unlabelled data, semi-supervised learning has been becoming promising, especially when it comes to training for deep architecture, such as deep belief network.

(3) Instead of having separate unsupervised and supervised components in the model, semi-supervised learning can construct models in which a generative model of either $P(\mathbf{v})$ or $P(\boldsymbol{v},\boldsymbol{u})$ shares parameters with a discriminative model of $P(\boldsymbol{u}/\boldsymbol{v})$. So that it is available to simultaneously remain the ability to extract input features and associated features.

REFERENCES

Abdel-Zaher, A. M., Eldeib, A. M. (2016). Breast cancer classification using deep belief networks. *Expert Systems with Applications*, 46, 139-144.

Chen, Y., Nasrabadi, N. M., Tran, T. D. (2013). Hyperspectral image classification via kernel sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 51(1), 217-231.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4), 303-314.

Gao, L., Li, J., Khodadadzadeh, M., Plaza, A., Zhang, B., He, Z., Yan, H. (2015). Subspace-based support vector machines for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 12(2), 349-353.

Huang, G. B., Chen, L., Siew, C. K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17(4), 879-892.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8), 1771-1800.

Hinton, G. (2010). A practical guide to training restricted Boltzmann machines. *Momentum*, 9(1), 926.

Hinton, G. E., Osindero, S., Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.

Hinton, G., & Salakhutdinov, R. (2011). Discovering binary codes for documents by learning deep generative models. *Topics in Cognitive Science*, 3(1), 74-91.

Jagannathan, S., Lewis, F. L. (1996). Identification of nonlinear dynamical systems using multilayered neural networks. *Automatica*, 32(12), 1707-1712.

Jagtap, J., Kokare, M. (2016). Human age classification using facial skin aging features and artificial neural network. *Cognitive Systems Research*, 40, 116-128.

Larochelle, H., Bengio, Y. (2008). Classification using discriminative restricted Boltzmann machines. *In Proceedings of the 25th international conference on Machine learning* (pp. 536-543). ACM.

Le Roux, N., Bengio, Y. (2008). Representational power of restricted Boltzmann machines and deep belief networks. *Neural computation*, 20(6), 1631-1649.

Lin, C. J., Chung, I. F., Chen, C. H. (2007). An entropy-based quantum neuro-fuzzy inference system for classification applications. *Neurocomputing*, 70(13), 2502-2516.

Liu, Y., Zhou, S., & Chen, Q. (2011). Discriminative deep belief networks for visual data classification. *Pattern Recognition*, 44(10), 2287-2296.

Shekofteh, Y., Almasganj, F., & Daliri, A. (2015). MLP-based isolated phoneme classification using likelihood features extracted from reconstructed phase space. *Engineering Applications of Artificial Intelligence*, 44, 1-9.

Shi, J., Zhou, S., Liu, X., Zhang, Q., Lu, M., Wang, T. (2016). Stacked deep polynomial network based representation learning for tumor classification with small ultrasound image dataset. *Neurocomputing*, 194, 87-94.

Tang, B., Liu, X., Lei, J., Song, M., Tao, D., Sun, S., & Dong, F. (2016). DeepChart: Combining deep convolutional networks and deep belief networks in chart classification. *Signal Processing*, 124, 156-161.

Yuan, H., Tang, Y. Y., Lu, Y., Yang, L., & Luo, H. (2014). Hyperspectral image classification based on regularized sparse representation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6), 2174-2182.

Zhong, Y., Ma, A., Zhang, L. (2014). An adaptive memetic fuzzy clustering algorithm with spatial information for remote sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(4), 1235-1248.