Information Sciences 463–464 (2018) 294–306

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A new deep belief network based on RBM with glial chains

Zhiqiang Geng[a,b], Zhongkun Li[a,b], Yongming Han[a,b,*]

[a] Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang, Guizhou 550025, China
[b] College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China

ARTICLE INFO

Article history:
Received 17 November 2017
Revised 8 May 2018
Accepted 17 June 2018
Available online 24 June 2018

Keywords:
Deep belief network
Glial chains
Unsupervised learning
Feature extraction
Image classification

ABSTRACT

A deep belief network (DBN) is an unsupervised learning method that is widely used to build hierarchical structures and learn feature representations based on unlabeled data; however, it is limited in feature acquisition because no interconnections occur between neurons in the same layer. Moreover, information shared between different neurons may be ignored, so deep correlative features of the data cannot be recognized. Inspired by the functions of glial cells in the neural network of the human brain, this paper proposes a variation of DBN based on Restricted Boltzmann Machines (RBMs) with glial chains. Additionally, an improved greedy layer-wise learning algorithm for the new DBN is applied to enhance learning accuracy and extract more information from the data. Furthermore, to ensure the glial effect is maintained, the glial chain is always contained in the supervised fine-tuning process. The experimental results based on benchmark image datasets show that the proposed DBN model can acquire more abstract features and achieve a lower mean squared error (MSE) than traditional DBN and other learning algorithms.

1. Introduction

Recently, deep learning [3] has been one of the hot topics in machine learning, pattern recognition, feature extraction and data mining [20]. Specifically, there is increasing interest in using the learning approach to build hierarchical representations based on unlabeled data. Deep learning is a branch of machine learning that tries to model high-level abstractions from sample data using complex networks or multiple non-linear transformation structures. Neuroscience findings suggest that much of the richness of perception is conveyed implicitly, however, perception is also a complex inferential process used to understand essential features. Therefore, the layer-by-layer abstraction of thousands of neurons in the brain is crucial [17]. Object recognition tasks can be done using multi-level abstraction of inputs corresponding to different areas of the cortex. In machine learning research, imitating the hierarchical architecture of the human brain to achieve better learning performance has become an essential issue.

Restricted Boltzmann Machines(RBMs) can be stacked in several layers, and the Deep Belief Network (DBN) [22] or Deep Boltzmann Machine (DBM) can be obtained based on the interactions between the layers [28]. DBN is a generative model with multiple densely connected layers of non-linear latent variables [11]. Because learning one layer at a time is relatively straightforward, DBN is an unsupervised learning approach that can easily realize representations learning from unlabeled data.

[*] Corresponding author at: College of Information Science and Technology, 15 Beisanhuan east road, Chaoyang district, Beijing 100029, China
E-mail address: hanym@mail.buct.edu.cn (Y. Han).

https://doi.org/10.1016/j.ins.2018.06.043
0020-0255/© 2018 Elsevier Inc. All rights reserved.

In recent years, various kinds of DBN models have been proposed. Using complementary priors, Hinton et al. derived a fast and greedy algorithm that can learn deep and directed belief networks layer-by-layer [11]. This algorithm was used to initialize a learning procedure that fine-tuned weights using a contrastive version of the wake-sleep algorithm. Then, a generative model could be formed to classify digits better than discriminative learning algorithms. However, this model was designed for binary images and did not have a systematic way to deal with perceptual invariances. Bengio et al. studied this algorithm further and successfully extended it to cases where inputs were continuous values or the structure of the input distribution did not reveal enough about the variable, which was predicted in a supervised task [2]. Experiments suggest that a greedy layer-wise training strategy can help optimize deep networks but that it is also important to have an unsupervised component to train each layer. Therefore, three-way RBMs are used in many fields with great results [38].

DBN has been successfully applied in many fields. Lee et al. presented a hierarchical generative model called convolutional DBN (CDBN), which scaled to realistic sizes [16]. The key approach of this DBN was probabilistic max-pooling which shrunk the representations of higher layers in a probabilistically sound way. Because of this, this model was translation-invariant and could support efficient bottom-up and top-down probabilistic inference. Huang et al. used CDBN to learn and acquire features from high-resolution images for face recognition. A new local convolutional RBM model was used to obtain additional representations and hand-crafted image descriptors such as local binary patterns. Experiments in the literature showed that learning weights were not only necessary to obtain good multilayer representations, but also improved the method's robustness when choosing network parameters [12]. Furthermore, a model for facial expression recognition, called Boosted Deep Belief Network (BDBN), was proposed by iteratively performing three training stages [19]. A set of features effective for characterizing expression-related facial appearances can be learned and selected to build a boosted classifier in a statistical way. Based on CDBN, Tanase et al. proposed a new CDBN network architecture that computed high-level, local and spatially invariant features from PolSAR patches to create fully polarimetric multi-view F-SAR images [32]. The experiments proved that the proposed CDBN efficiently exploited the informational content in this special type of data to extract high-level, local and spatially invariant features. To improve recognition performance further, Roy et al. presented an approach using a word hypothesis rescoring scheme based on DBN [26]; the efficient discriminative feature of the DBN was combined with the sequence classifier based on recurrent neural networks. Mohamed et al. achieved better phone recognition on the TIMIT dataset using DBNs instead of Gaussian mixture models [21]. The networks were first pretrained without discriminative information and then fine-tuned using back propagation. To fully utilize the generative nature of DBN, Kang et al. modeled speech parameters, including spectrum and F0, and generated these parameters simultaneously using DBN for speech synthesis [15]. The DBN model spectrum is better than the Hidden Markov Model (HMM) [23] and has less distortion. Tu et al. proposed graph-induced RBM to encode domain-specific knowledge [33]. Comprehensive experiments proved that the proposed model can prevent overfitting and improve word group coherence with or without transferring prior knowledge, document classification and word embedding. Wang et al. proposed a semi-supervised DBN (SSDBN) based on a semi-supervised restricted Boltzmann machine to short the gap and improve the accuracy of classification [36]. The experimental results show that SSDBN outperforms traditional DBN and other models with respect to classification. Qiao et al. proposed a self-organizing DBN (SODBN) with growing and pruning algorithms for nonlinear system modeling [25]. The SODBN could automatically determine its structure using growing and pruning algorithms instead of artificial experience. Zand et al. proposed a resistive DBN (R-DBN) architecture using the physics of nanomagnets to provide a natural hardware representation for individual probabilistic neurons [39]. Zhang et al. proposed a multi-objective DBN ensemble (MODBNE) method employed a multi-objective evolutionary algorithm integrating the traditional DBN training technique to evolve multiple DBNs simultaneously subject to two conflicting objectives [40]. Wang et al. proposed a cycle DBN model to classify multivariate time series (MTS) data [37]. In the authorship verification field, Brocardo et al. used Gaussian units in the visible layer to model real-valued data based on Gaussian-Bernoulli DBN [4]. Meanwhile, DBN is widely used in other fields, such as human behavior analysis [24], scene recognition [7], speech recognition [27,35], facial image reconstruction [34] and Bearing Fault Diagnosis [5,31].

Although DBN has shown better results in many application fields, DBN without constraints on hidden layers may produce unstructured weight patterns. Therefore, some solutions based on neuroscience research are introduced as follows. In recent neuroscience studies, glial cells have come to the center stage to help understand the working scheme of the human brain [9]. Glial cells can transmit signals using ions, such as Ca^{2+}, glutamic acid (Glu) and adenosine triphosphate (ATP). Among these ions, Ca^{2+} can change the membrane potential of neurons and the states of neighboring glial cells. Some researchers have recognized how biological glial cells work in the brain and applied that behavior to artificial neural networks [14]. The proposed Multilayer Perceptron (MLP) method had many glial cells in a hidden layer. The glial cells were connected with neurons and excited by the outputs of those neurons; excited glial cells could send signals to neighboring glial cells. The proposed MLP aimed to obtain the relationship between neuron positions in the hidden layer. Therefore, the obtained information could have a good influence on the learning process. Since there are no connections between units in the same layer in DBN, the relationship information in the same layer can be obtained by adding chains of glial cells. Geng et al. proposed an improved DBN with glial chains [8], but the fine-tuning process did not use glial chains, which may cause the effect of glial chains to disappear.

To ensure interconnection among a layer of neurons and obtain effective correlation features among different dimensions of the same data, a variant structure of DBN based on RBMs connected with glial chains is proposed. Additionally, an improved greedy layer-wise learning algorithm for the new DBN is used to enhance learning accuracy and extract more information from the data. During the RBM training process and the supervised BP fine-tuning training process, glial cell

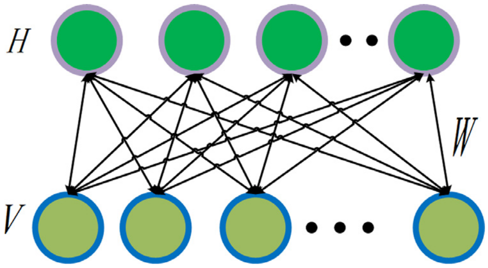

Fig. 1. Structure of RBM.

function can adjust the activation probability of hidden units and send pulse signals to other glial cells. Moreover, the hidden units in the same layer are not isolated during the unsupervised pre-training process or the supervised BP process, before their correlation features are obtained. Furthermore, to better discriminate the states of the neurons, the proposed model ensures that the states of the neurons during the entire training process are affected not only by the states of the neurons in the last layer but also by the states of other neurons in the same layer. The experimental results based on benchmark image datasets show that the proposed DBN model can acquire more abstract features and achieve a lower mean squared error (MSE) than traditional DBN and other learning algorithms.

This paper is divided into five sections. Section 1 mainly introduces the current research background and research status quo. The related principles of RBM and DBN and the contrastive divergence (CD) algorithm process are described in Section 2. Section 3 describes the specific process of the improved DBN algorithm with the icon and pseudo-code. The experimental results from further analysis and verification of the proposed method through the use of datasets and specific experiments are shown in Section 4. The conclusion is given in Section 5.

2. DBN and RBM

In this section, the principles and relationship between DBN and RBM are elaborated. DBN is a probabilistic model consisting of several hidden layers. Each hidden layer can get higher-order features from the lower layer through learning. Additionally, DBN can be built by stacking several layers of models, called RBM.

2.1. RBM

RBM is a bipartite, undirected graphic model [6]. As shown in Fig. 1, RBM consists of a visible layer and a hidden layer. The visible units of dimension V and hidden units of dimension H are connected by a symmetrical weight matrix $[\boldsymbol{W}]_{V*H}$. It is worth noting that there is no connection among the nodes of the hidden layer or among the nodes of the visible layer.

The joint distribution over the visible and hidden units can be defined as

$$P(\mathbf{v},\mathbf{h}) = \frac{e^{-Egy(\mathbf{v},\mathbf{h})}}{\mathrm{T}} \tag{1}$$

$$\mathrm{T} = \sum_{\mathrm{v}}^{\mathrm{V}}\sum_{\mathrm{h}}^{\mathrm{H}} e^{-Egy(\mathbf{v},\mathbf{h})} \tag{2}$$

Where **v** is the vector of visible units, **h** is the vector of hidden units,and T is a normalization constant defined as the sum of $\mathrm{e}^{-Egy(\mathbf{v},\mathbf{h})}$ over all possible values of **v, h.**

The weights and biases of RBM determine the energy function of a joint distribution of hidden and visible units, $Egy(\mathbf{v},$ $\mathbf{h})$. If the visible units are binary-valued, the energy function is defined as:

$$Egy(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{V}\sum_{j=1}^{H} W_{ij}v_ih_j - \sum_{i=1}^{V} a_iv_i - \sum_{j=1}^{H} b_jh_j \tag{3}$$

Where a_i and b_j are the biases of the visible and hidden units, respectively, and W_{ij} is the weight between the visible and hidden units. If the visible units are real-valued, the energy function can be defined as

$$Egy(\mathbf{v},\mathbf{h}) = \frac{1}{2}\sum_{i=1}^{V} v_i^2 - \sum_{i=1}^{V}\sum_{j=1}^{H} W_{ij}v_ih_j - \sum_{i=1}^{V} a_iv_i - \sum_{j=1}^{H} b_jh_j \tag{4}$$

From the energy function, it is clear that, given the states of the visible units, the hidden units are independent of each other; this the same for the visible units. Defined with conditional probability distributions, for the hidden layer, the binary

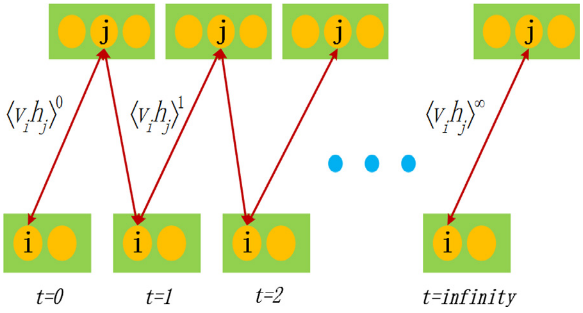
j
j
j
j
$\langle v_i h_j \rangle^0$
$\langle v_i h_j \rangle^1$
$\langle v_i h_j \rangle^\infty$
i
i
i
i
t=0
t=1
t=2
t=infinity

Fig. 2. Multistep sampling in the CD algorithm.

state of each hidden unit h_j is set to 1 with probability:

$$p(h_j = 1|\mathbf{v}) = \boldsymbol{sigma}\left(\sum_i^V W_{ij} v_i + b_j\right) \tag{5}$$

where $\boldsymbol{sigma}(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. Similarly, if the visible layer is binary-valued, the visible units are conditioned on the hidden layer and the state of each visible unit v_i is set to 1 with probability:

$$p(v_i = 1|\mathbf{h}) = \boldsymbol{sigma}\left(\sum_j^H W_{ij} h_j + a_i\right) \tag{6}$$

If the visible layer is real-valued, the visible units are independent Gaussians with diagonal covariance:

$$p(v_i|\mathbf{h}) = \boldsymbol{N}\left(\sum_j^H W_{ij} h_j + a_i, 1\right) \tag{7}$$

where $\boldsymbol{N}(*, *)$ is a Gaussian distribution function.

RBM is a generative model, and its parameters can be optimized using stochastic gradient descent based on the Maximum log-likelihood of the training data. Since the computation of the exact gradient is difficult, an approximation method is often used, the most common of which is the CD algorithm.

2.2. Contrastive divergence algorithm

Based on the gradient of the log likelihood $\log P(v)$, the update rule for the weights in RBM can be easily derived as shown in Eq. (8):

$$\Delta w_{ij} = \boldsymbol{Et}_{real}(v_i h_j) - \boldsymbol{Et}_{output}(v_i h_j) \tag{8}$$

where $\boldsymbol{Et}_{real}(v_i h_j)$ is the observed expectation of the training data and $\boldsymbol{Et}_{output}(v_i h_j)$ is the expectation under the distribution defined by the training model [21]. It is difficult to compute $\boldsymbol{Et}_{output}(v_i h_j)$, so the CD method [10] is commonly used to approximate it. By conducting the sampling process on the data for one or more steps, the expectations will be updated. For one step sampling(CD-1), this procedure can be stated as:

(1) Get $\boldsymbol{v_0}$ from the training data.
(2) Calculate sample $\boldsymbol{h_0} \sim p(\mathbf{h}|\boldsymbol{v_0})$.
(3) Calculate sample $\boldsymbol{v_1} \sim p(\mathbf{v}|\boldsymbol{h_0})$.
(4) Calculate sample $\boldsymbol{h_1} \sim p(\mathbf{h}|\boldsymbol{v_1})$.

Each sampling step means the states of hidden units (or visible units) are determined by a conditional probability given the fixed states of visible units(or hidden units). For example, h_j is randomly chosen to be 1 (or 0) with probability $\boldsymbol{sigma}(\sum_i^V W_{ij} v_i + b_j)$; similarly v_i is randomly chosen to be 1 (or 0) with probability $\boldsymbol{sigma}(\sum_j^H W_{ij} h_j + a_i)$.

Details of the CD algorithm will be described in Section 3.2. For more than one steps, the sampling process can be depicted in Fig. 2.

Several layers of RBMs can be stacked to build a DBN. Becausethe hidden layer of an RBM serves as the visible layer of the next RBM, each layer can capture high-order correlations between the activities of hidden features in the layer below. DBN can be trained using a repeated greedy layer-by-layer algorithm [18]. Meanwhile, the bottom layer is first trained with CD on the training data. Then, the parameters are frozen and the hidden values are inferred as the inputs of the next layer. After several repetitions, a deep architecture is learned.

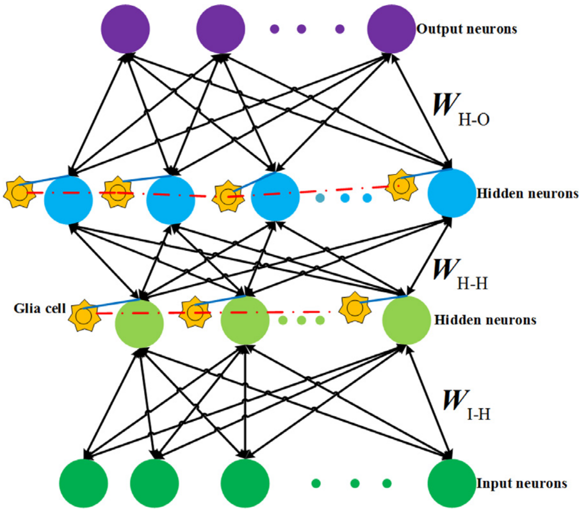

Fig. 3. The improved DBN construction with glial chains.

3. RBM and DBN with glial chains

The principles and the detailed process of RBM and DBN with glial chains are described in this section. Glial cells are a type of nervous cell in the human brain. They can transmit signals to neurons and other glial cells. Researchers have noticed that the features of glia can be applied to the learning process of artificial neural networks. Ikuta et.al. proposed an MLP with pulse glial chains [13]. During the training process, the glia generate pulses and propagate them in the neural network. From experimental results, this model has shown that it has better performance than conventional MLP.

Similarly, DBN can be regarded as part of an artificial neural network called the pretrained deep neural network (DNN) [1]. This type of model uses unsupervised pretraining to facilitate the subsequent discriminative fine-tuning process. As there are no connections between units of the same layer, relationships within a layer may be ignored. Inspired by the model with glial chains mentioned above, we believe that the glial cells can benefit from RBM training. Therefore, we simplify the definition of glial chain and adapt it to the structure of RBM. The new structure of DBN with glial chains is shown in Fig. 3.

Apart from the three layers of RBM, there is another set of glial cells represented by a star graph connected as a chain. Each glial cell also has a connection to a hidden layer unit corresponding to its position. With these connections, glial cells can transmit signals to units of the RBM and other glial cells.

3.1. Updating rule of the hidden units' output

In the training process of the RBM, the output of the connected hidden unit is adjusted by the activated glial cell, and a pulse is generated to other glia. For example, if the output of hidden unit h_1 is larger than the given threshold, glia g_1 will be activated and a pulse will be generated to its neighboring glia g_2 .When the pulse reaches g_2, even if the output of h_2 doesn't reach the threshold, g_2 is still activated, a second pulse is generated, and the first pulse will continue to propagate. In our implementation, to simplify computation, the pulses are propagated in the same direction(from the first glia to the last one).

The updating rule of the hidden units' output can be defined as:

$$h_j = \boldsymbol{sigma}(\widehat{h}_j + \mu * f_j) \tag{9}$$

where h_j is the updated output, *sigma* is the sigmoid function, f_j is the glial effect, and μ is the weight of the glial effect. in particular, $\widehat{h}_j$ is the original input to the hidden unit, which can be calculated as:

$$\widehat{h}_j = \sum_i W_{ij} v_i + b_j \tag{10}$$

where W_{ij} is the weight connected to the hidden unit, v_i is the state of the visible unit, and b_j is the bias of the hidden layer. We use this probability directly instead of sampling a binary value as the output, because it could reduce sampling noise and realize faster learning [13].

As for the glial effect, f_j can be defined as:

$$f_j(s) = \begin{cases} 1, (\widehat{h}_j > \Omega \vee f_{j-1}(s-1) = 1) \wedge (s_j < S), \\ \mu f_j(s-1), others \end{cases} \tag{11}$$

where Ω is the given threshold, S is an inactive period after activation, and μ is an attenuation factor. More specifically, in this definition, the pulse generated by the activated glia transmits to the next glia in a unit of time. A glia will be activated if the output of the connected hidden unit reaches the limit or if the previous glia transmits a pulse to it and its last activation is a certain time (S) ago. If activated, the glia will continue to transmit the signal; otherwise, its glial effect will be attenuated.

3.2. RBM and DBN learning algorithm with glial chains

Owing to the added glial effect, the RBM learning algorithm will be improved. Each time outputs from the hidden layer are computed, they are adjusted by the glial chain according to the previous state and the glial effect for the next state is saved. Pseudocode of the proposed algorithm is shown in the following. The improved DBN algorithm with glial chains is shown as follows.

```
Input: sample vector $\mathbf{x}_0$, learning rate of Contrastive Divergence $\eta$, glial effect vector $\boldsymbol{f}$
Output: Weight matrix $[\boldsymbol{W}]_{V*H}$, biases vector of hidden units $\mathbf{b}$, biases vector of visible units $\boldsymbol{a}$
Training(Iterate k times depending on the value of k in CD-k):
•$\mathbf{v_0} = \mathbf{x_0}$
foreach hidden unit$j : [0, H)$ do
    •calculate posterior probability $p(h_{0j} = 1|\mathbf{v_0}) = sigma(\sum_{i}^{V} W_{ij}v_{0i} + b_j)$
    •sample $h_{0j}$ according to $p(h_{0j}|\mathbf{v_0})$
end
foreach visible unit $i : [0, V)$ do
    •calculate posterior probability $p(v_{1i} = 1|\mathbf{h_0}) = sigma(\sum_{j}^{H} W_{ij}h_{0j} + a_i)$
    •sample $v_{1i}$ according to $p(v_{1i}|\mathbf{h_0})$
end
    •compute $\widehat{h}_{1j} = \sum_{i}^{V} W_{ij}v_{1i} + b_j$
    •update the glial vector $\boldsymbol{f}$ by one step according to Eq. (11) and save the glial states for the next iteration
foreach hidden unit $j : [0, H)$ do
    •compute $h_{1j} = sigma(\widehat{h}_{1j} + \mu * f_j)$
end
    •update parameters:
    $[\boldsymbol{W}]_{V*H} = [\boldsymbol{W}]_{V*H} + \eta(p(\mathbf{h_0} = 1|\mathbf{v_0})\mathbf{v_0}^T - p(\mathbf{h_1} = 1|\mathbf{v_1})\mathbf{v_1}^T)$
    $\mathbf{b}$=$\mathbf{b} + \eta(p(\mathbf{h_0} = 1|\mathbf{v_0}) - p(\mathbf{h_1} = 1|\mathbf{v_1}))$
    $\boldsymbol{a} = \boldsymbol{a} + \eta(\mathbf{v_0} - \mathbf{v_1})$
end training
```

A deep belief network consists of several RBMs and could be trained by a greedy layer-by-layer algorithm. When one of the RBMs has been trained, the learned parameters are kept and the outputs of the hidden layer can be used as inputs to the next RBM. The next RBM will also be trained using the CD algorithm.

3.3. Supervised BP neural network fine-tuning process with glial chains

After the unsupervised pretraining process when we use the improved RBM model, we add the labels of the training data to fine-tune the weights of the neural networks. In this process, to ensure correlation between the same layers of neurons, the glial chains are kept for supervised training. In this way, better results can be obtained.

The procedure and flow chart of the improved DBN are shown in detail in Fig. 4.

4. Experiments

To evaluate the learning performance of the proposed model, we analyze experiments on four image classification datasets: the Columbia University Image Library (Caltech 101) dataset, COIL-20 dataset, MNIST dataset and Optical Character Recognition (OCR) dataset. The results of the proposed model are compared with those of some other algorithms (conventional RBM, Sparse Auto-encoder [29], BP Neural Network [41]). To increase the learning rate, we adopt the mini-batch learning method. The datasets are divided into several batches and the weights are updated after each mini-batch computation. In the experiment, we normalize the dataset of the pixel matrix and the labels of these data sets. Finally, the square of the difference between the neural network output and the normalized label is the actual error, the MSE. When testing the testing data, the same standard is used. The formula for MSE is:

$$MSE = \frac{1}{n}\sum_{i}(output_i - y_i)_i^2$$

Where $output_i$ is the model output for the i-th data and y_i is its label.

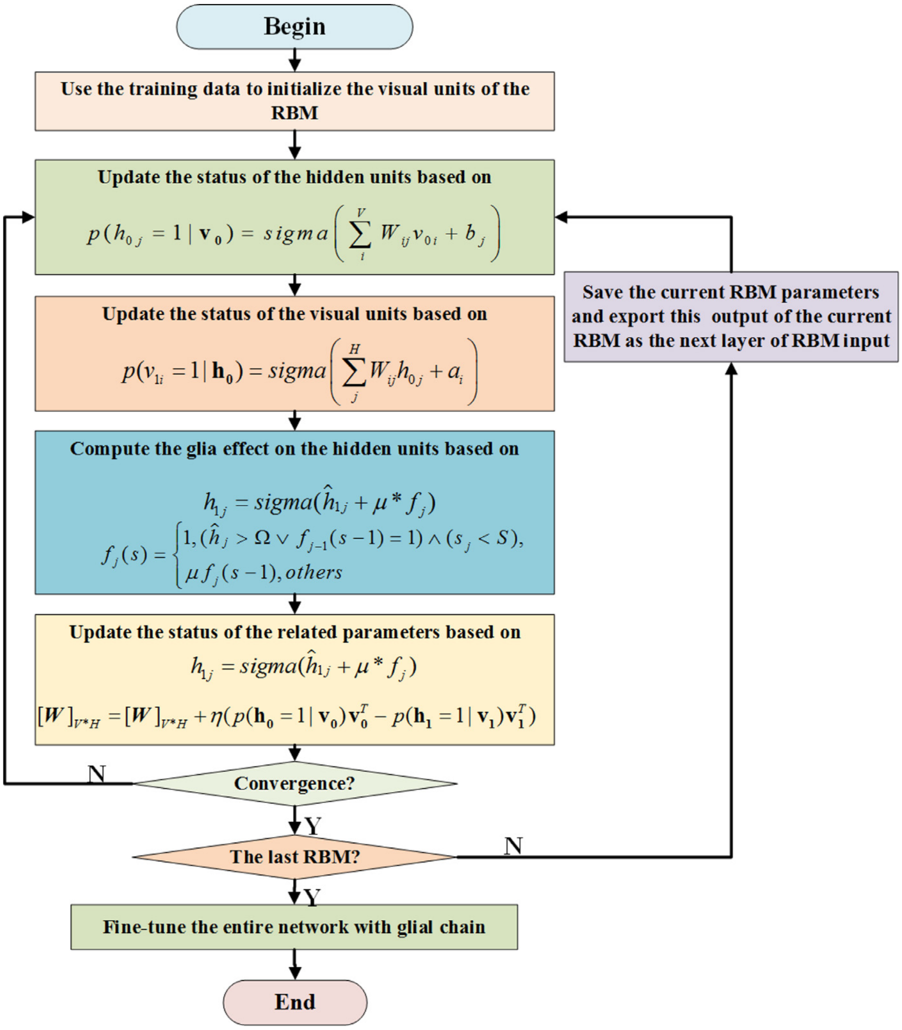
Begin
Use the training data to initialize the visual units of the RBM
Update the status of the hidden units based on
$p(h_{0j}=1|\mathbf{v}_0)=sigma\left(\sum_i^V W_{ij}v_{0i}+b_j\right)$
Save the current RBM parameters and export this output of the current RBM as the next layer of RBM input
Update the status of the visual units based on
$p(v_{1i}=1|\mathbf{h}_0)=sigma\left(\sum_j^H W_{ij}h_{0j}+a_i\right)$
Compute the glia effect on the hidden units based on
$h_{1j}=sigma(\hat{h}_{1j}+\mu*f_j)$
$f_j(s)=\begin{cases}1,(\hat{h}_j>\Omega\vee f_{j-1}(s-1)=1)\wedge(s_j<S),\\ \mu f_j(s-1), others\end{cases}$
Update the status of the related parameters based on
$h_{1j}=sigma(\hat{h}_{1j}+\mu*f_j)$
$[W]_{V*H}=[W]_{V*H}+\eta(p(\mathbf{h}_0=1|\mathbf{v}_0)\mathbf{v}_0^T-p(\mathbf{h}_1=1|\mathbf{v}_1)\mathbf{v}_1^T)$
N
Convergence?
Y
The last RBM?
N
Y
Fine-tune the entire network with glial chain
End

Fig. 4. Training flow chart of the improved DBN.

4.1. COIL-20 dataset

The COIL-20 dataset is a large dataset of many pictures of objects belonging to 20 categories; it contains 1440 images in total. Each image, from which the background has been removed, contains a specific object; 1200 images are used to train models, and 240 images are used to test the model.

Firstly, we trained RBM and RBM with glial chains with 4096 visible units and 100 hidden units. The learned bases of these two models are shown in Fig. 5.

It can be seen in Fig. 5 that the bases learned by RBM have more blob detectors but less future edges, most of which are not clearly recognized. As for RBM with glial chain, the bases seem more discriminative; most have clear and distinct edge features, only a few are blob detectors, and the local characteristics and differentiation of objects are obvious and clear.

Then we train RBM, RBM with glial chain, Sparse Auto-encoder and Neural Network (NN) with different numbers of hidden units. To compare the classification MSE, the number of hidden units is increased from 200 to 500 to test the aforementioned models on the testing images. These results are shown in Table 1.

The results suggest that RBM with glial chain has better classification performance than conventional RBM and other algorithms. As the number of hidden units increased, the MSE of RBM with glial chain, which was always the lowest, decreased. Therefore, features learned by the proposed model are more effective and discriminative for classification.

To further study the learning performance of the hierarchical structure, DBN and DBN with glial chains are trained on the COIL-20 dataset. The structures of these two models are the same, with 500 units in the first hidden layer and 200 units in the second hidden layer, and common parameters (such as the learning rate and momentum) are also set to be the same. The average reconstruction MSE of training and testing on image classification is shown in Table 2.

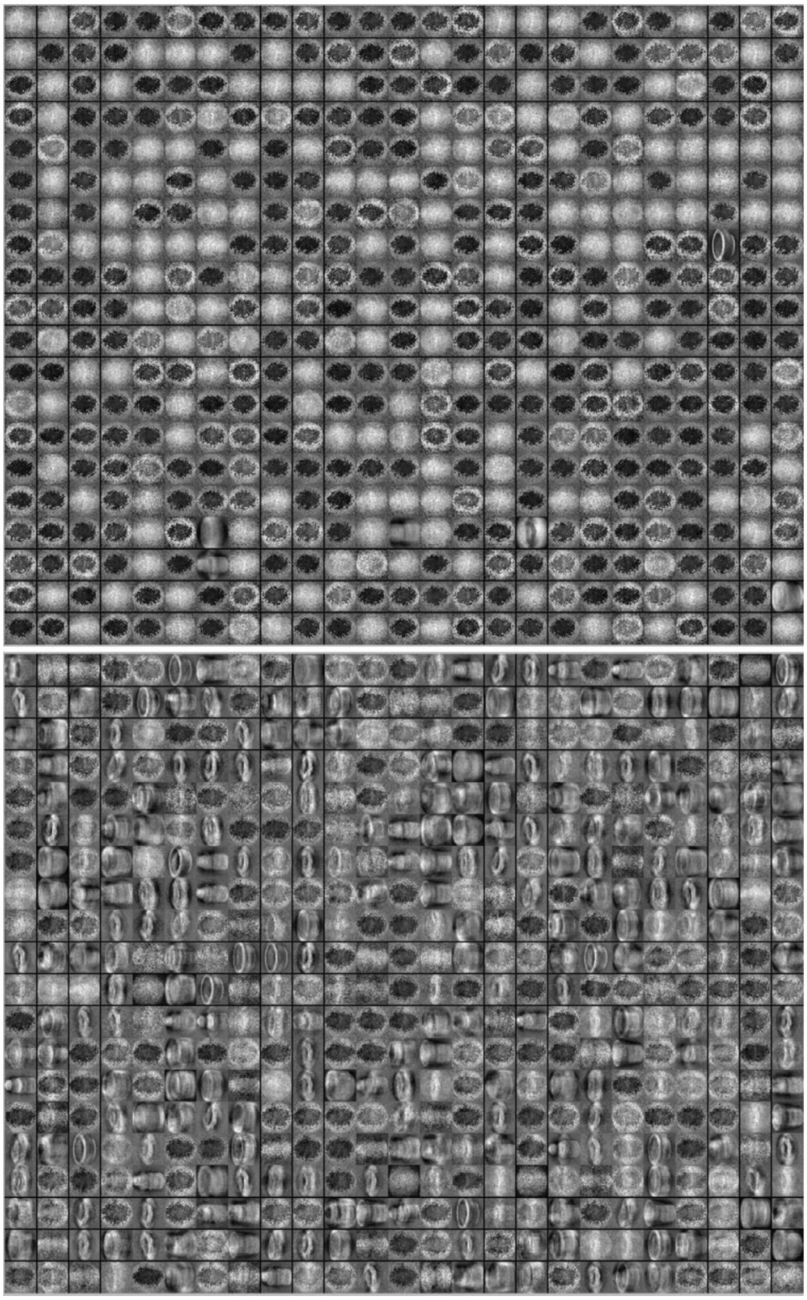

Fig. 5. Visualization of features learned by RBM and improved RBM.

Table 1
Testing results of different models on the COIL-20 dataset.

Models	200 hidden units		350 hidden units		500 hidden units	
	Testing MSE(%)	Convergence time(s)	Testing MSE(%)	Convergence time(s)	Testing MSE(%)	Convergence time(s)
RBM	1.8957	12.958	1.4624	19.966	1.1108	27.433
Sparse Auto-encoder	1.1862	19.827	1.4554	30.444	1.139	42.365
BP neural network	2.3302	10.877	1.561	17.595	1.5522	24.546
RBM with glial chain	**1.1476**	**13.434**	**0.9068**	**20.105**	**0.56674**	**27.486**

Table 2
Training and testing MSEs of DBN and improved DBN on the COIL-20 dataset.

Models	Training MSE (%)	Testing MSE (%)
DBN	0.7521	1.7446
DBN with glial chains	**0.4411**	**0.9099**

Table 3
Comparison of DBN and other models' best results on the COIL-20 dataset.

Models	Testing MSE (%)
3-layer NN, 500 + 300 hidden units, softmax, weight decay	1.1486
Support Vector Machine, Gaussian kernel	2.50
DBN, using RBMs pre-training a 4096-500-500-2 000 network	0.7631
DBN, using RBMs with glial chain pretraining a 4096-500-500-2 000 network	**0.6504**

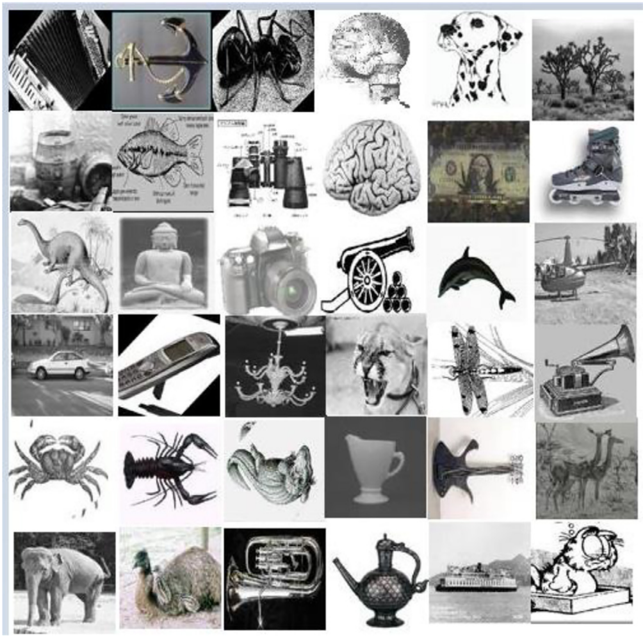

Fig. 6. Image samples from the Caltech 101 training dataset.

Based on Table 2, for hierarchical structures, the testing rate of DBN with glial chains is still lower. This further verifies our hypothesis that glial chains can improve the learning of a hierarchical structure. With glial chains, DBN takes into account the relationship of units within a hidden layer. Moreover, the glial cells can transmit signals to each other and adjust the outputs of hidden units.

In previous research, a 784-500-500-2000 deep network with the same numbers of glial cells connected to the hidden layers is trained to reach its best classification result. Pretraining data containing 1200 images are divided into several mini-batches. After pre-training, the network is fine-tuned using BP. Then, the final classification accuracy is taken as the basis for selecting parameters. It can be seen in Table 3 that DBN with glial chains outperforms the other listed models.

4.2. Caltech 101 dataset

The Caltech 101 dataset consists of 8677 3-channel color images in 101 classes, with dozens to hundreds of images in each class; it is divided into two parts: 6000 training samples and 2677 test samples. In each image, there is an object in one of the 101 classes (such as an accordion, anchor, etc.). The classes are mutually exclusive. Compared with the COIL-20 dataset, the Caltech 101 dataset is more complex for classifiers because the images are colored and of a relatively larger size; also, extracting features will be harder as the shapes of each class' objects are very different. To improve the training speed, according to the characteristics of the image, we convert each image to an 80 × 80 grayscale image. Some samples used for training are shown in Fig. 6.

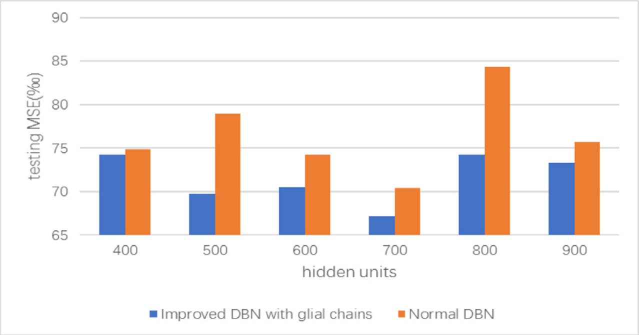

Fig. 7. Testing MSEs with different numbers of hidden units.

Table 4
Training and testing MSEs of DBN and improved DBN on the Caltech 101 dataset.

Models	Training MSE (%)	Testing MSE (%)
DBN	2.8454	6.1875
DBN with glial chains	**2.5060**	**6.0405**

Table 5
Testing MSEs of DBN and improved DBN on the MNIST dataset.

Models	Testing MSE(%)					
	200 hidden units	300 hidden units	400 hidden units	500 hidden units	500+300 hidden units	300+200 hidden units
DBN	1.1771	0.9486	0.8917	0.83789	0.67938	0.71843
DBN with glial chains	**1.0998**	**0.93212**	**0.88228**	**0.82022**	**0.66799**	**0.70799**

Similar to the previous experiments, we train RBM and RBM with glial chain with 6400 visible units (each pixel of the 80 × 80 gray image) and gradually increase the number of hidden units from 400 to 900. Classification MSEs are shown in Fig. 7.

Based on Fig. 7, the MSE of RBM with glial chain is still lower than RBM. Although the input data become larger and more complicated, RBM with glial chains is still able to learn better features of the images. Thus, during the learning procedure, glial chains can help change the weights to reduce MSE.

Then, a two-hidden-layer structure with 500 and 200 hidden units is built. The training and testing MSE rates are shown in Table 4. These results show that DBN with glial chains achieves lower training MSEs and better classification accuracy.

4.3. MNIST dataset

The MNIST dataset is a large dataset of handwritten digits that is widely used for training and testing in the machine learning field; it contains 60,000 training images and 10,000 testing images. Each image is a handwritten digit from 0–9 with 28 × 28 pixels; the labels of these data are converted to the corresponding numbers and then are normalized.

Firstly, we pre-trained the DBN and proposed DBN with 784 visible units and different numbers of hidden units. The number of hidden units is increased from 200 to 500 to test the DBN and proposed DBN on the testing images. Furthermore, the hidden units are divided into two layers; the number in the first layer and the second layer in one hidden unit is 500 and 300 (500 + 300), respectively. Meanwhile, the number in the first layer and the second layer in another hidden unit is 300 and 200 (300 + 200), respectively. The classification MSEs are shown in Table 5.

It can be seen from Table 5 that the proposed DBN has better classification performance than the conventional DBN. For one hidden layer, as the number of the hidden units increased, the MSE of the proposed DBN, which is always lower, decreased. For two hidden layers, the proposed DBN also achieves the lowest MSE. When the proposed DBN has two hidden layers with 500 + 300 units, the testing MSE is 0.66799, which is the lowest in Table 5.

4.4. Parameter influence analysis

There are three parameters to be determined for DBN with glial chains: glial effect weight α, attenuation factor β and the threshold θ. In this section, we will discuss how these parameters affect the image classification accuracy based on experimental results obtained using the OCR dataset, which is a handwritten lowercase dataset with a size of 16 × 8. A total of 30,000 data are used as a training set, and the remaining data are used as a test set.

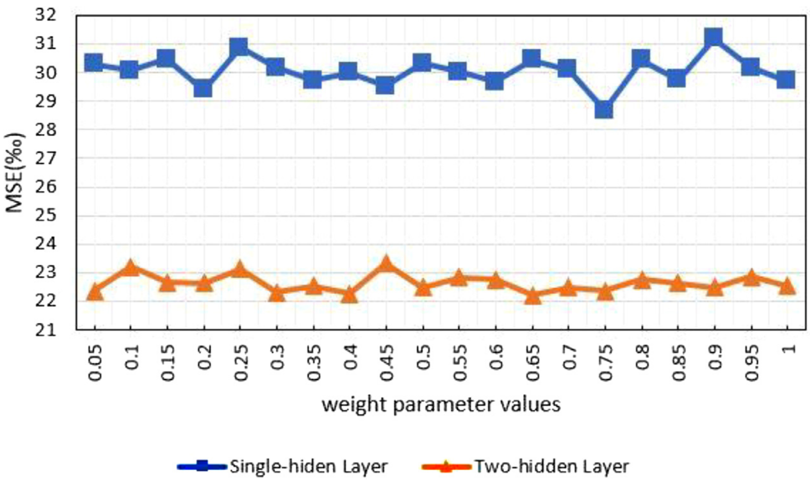

Fig. 8. Testing MSE under different weight parameters for glia classification.

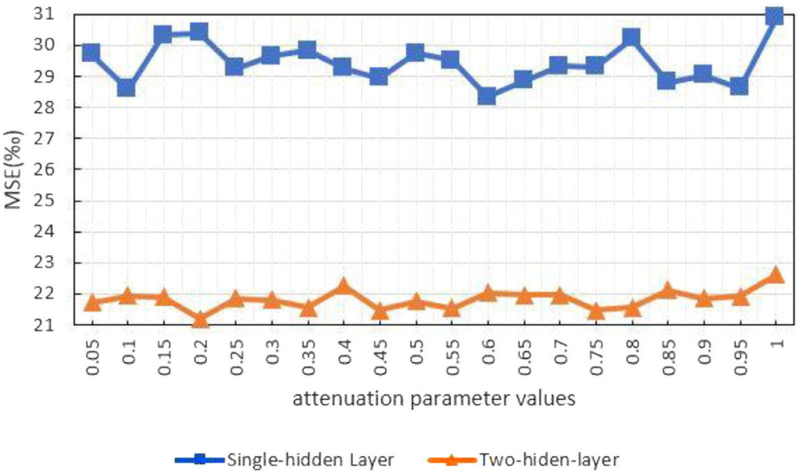

Fig. 9. Testing MSE with different values of attenuation factors.

DBN with glial chains is trained to set the three parameters at different values. The network structure is fixed as being trained with one hidden layer (500 hidden units) and two hidden layers (500 hidden units in the first hidden layer, 300 in the second).

Testing MSEs for different values of the glial effect weight α for the OCR dataset are shown in Fig. 8.

It can be seen in Fig. 8 that when the glial effect weight ranges from 0.05 to 1, the testing MSE fluctuates; however, for one-hidden-layer and two-hidden-layer models, the highest MSE occurs when the weight is too small (0.05) or too large (0.90). The lowest MSE occurs when the weight is 0.50 and 0.75. Therefore, we suppose that it is feasible to obtain the best result when the weight is between 0.50 and 0.75. Meanwhile, when the weight is set to 0.7, MSE seems to be average.

The MSEs obtained using different values of attenuation factor β on the OCR dataset are shown in Fig. 9. In this case, the highest MSE occurs at the value of 1.00 for both the one-hidden-layer model and the two-hidden-layer model. The lowest MSE is different for the one-hidden-layer and two-hidden-layer models; the lowest MSE of the one-hidden-layer model occurs between 0.15 and 0.35, but the lowest MSE of the two-hidden-layer model occurs between 0.85 and 0.95. We can search the best result using these intervals. It is noticeable that 0.5 is an average point for MSE.

Testing MSEs with different values of threshold θ are shown in Fig. 10. For the one-hidden-layer model, the lowest and highest MSEs occur at 0.1 and 0.75, respectively. Similarly, for the two-hidden-layer model, the lowest and highest values occur at 0.15 and 0.75, respectively. which are close to those of the one-hidden-layer model. However, the lowest MSEs occur when the threshold is set to 0.1 and 0.4, and we find that when the set threshold is between 0.45 and 0.75, the MSEs have larger fluctuation. Therefore, it is feasible to set the threshold larger or smaller for models with more hidden layers.

From experiments on the OCR dataset, we can find a few patterns for the three parameters, but our work is limited as to find out how these parameters affect each other. There is still a lot of work to be done in further studies.

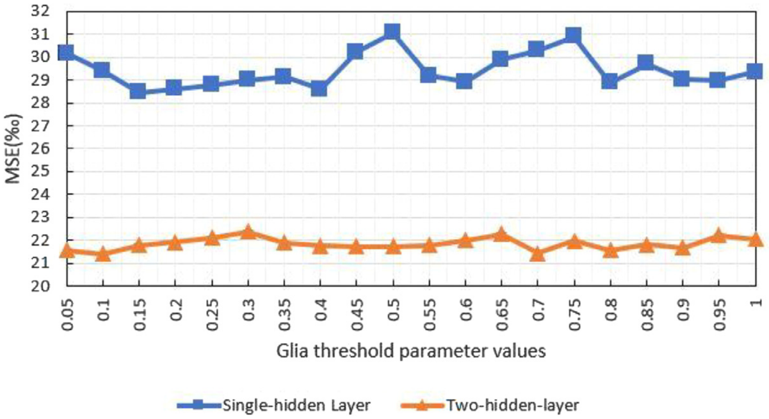

Fig. 10. Testing MSE with different values of the glia threshold.

In the experiment section, four simulation experiments were carried out. The first three simulation experiments compared models in many perspectives such as the visualization of features, the testing MSE with different numbers of hidden units and different neural network layers, wherein the proposed model proved to extract more efficient features and achieve better effects. In the fourth simulation experiment, by setting different parameters, the usability of the proposed method was verified.

5. Conclusions

A new DBN model based on biological interactions between glial cells and neurons in the human brain is proposed. The glial chains are added to the hidden layer of RBM. On the foundation of the proposed neural network structure, an improved greedy layer-wise learning algorithm is proposed to enhance learning accuracy and extract more information from the training data. When RBM is being trained, glial cells can transmit signals to other glial cells and adaptively adjust outputs of hidden layer units. With little additional computation, the modified DBN can be trained easily using an improved CD algorithm. The COIL-20, Caltech 101, MNIST and OCR datasets have been used to verify the effectiveness of the proposed model. Compared with other algorithms, DBN with glial chains can extract and learn features better for image classification tasks. Additionally, some key parameters are tested to see the effect of different values to provide some advice on selecting them to get the best learning results. Reducing running time and developing a self-organizing structure for DBN will be the focus of our future work. Meanwhile, to extract more effective features, glial cells are applied as encoders and decoders in the auto-encoders. Moreover, the glial cells are used in the generative network and the discriminator network in generative adversarial networks (GAN). Furthermore, the proposed DBN can be treated to a feature generator in natural language understanding [30].

Acknowledgment

This research was partly funded by National Natural Science Foundation of China (61673046 and 61603025), the Natural Science Foundation of Beijing, China (4162045), the National Key Research and Development Program of China (2017YFC1601800) and Science and Technology Major Project of Guizhou Province (Guizhou Branch [2018]3002).

References

[1] Y. Bengio, in: Learning Deep Architecture for AI, 2, Foundations & Trends in Machine Learning, 2009, pp. 1–127.
[2] Y. Bengio, P. Lamblin, P. Dan, H. Larochelle, U.D. Montréal, M. Québec, Greedy layer-wise training of deep networks, in: Advances in Neural Information Processing Systems, MIT Press, 2006, pp. 153–160.
[3] P.P. Brahma, D. Wu, Y. She, Why deep learning works: a manifold disentanglement perspective, IEEE Trans. Neural Netw. Learn. Syst. 27 (10) (2016) 1997–2008.
[4] M.L. Brocardo, I. Traore, I. Woungang, M.S. Obaidat, Authorship verification using deep belief network systems, Int. J. Commun. Syst. 30 (12) (2017) e3259.
[5] Z. Chen, W. Li, Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network, IEEE Trans. Instrum. Meas. 66 (7) (2017) 1693–1702.
[6] A. Fischer, C. Igel, An introduction to restricted boltzmann machines, Lect. Notes Comput. Sci. (2012) 14–36.
[7] J. Gao, J. Yang, G. Wang, M. Li, A novel feature extraction method for scene recognition based on centered convolutional restricted Boltzmann machines, Neurocomputing 214 (C) (2016) 708–717.
[8] Z.Q. Geng, Y.K. Zhang, An improved deep belief network inspired by glia chains, Acta Autom. Sin. 42 (6) (2016) 943–952.
[9] P.G. Haydon, GLIA: listening and talking to the synapse, Nat. Rev. Neurosci. 2 (3) (2001) 185–193.
[10] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (8) (2014) 1771–1800.
[11] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2014) 1527–1554.

[12] G.B. Huang, H. Lee, E. Learned-Miller, Learning hierarchical representations for face verification with convolutional deep belief networks, IEEE Conf. Comput. Vis. Pattern Recognit. 157 (10) (2012) 2518–2525.

[13] C. Ikuta, Y. Uwate, Y. Nishio, Multi-layer perceptron with positive and negative pulse glial chain for solving two-spirals problem, Int. Joint Conf. Neural Netw. IEEE (2012) 1–6.

[14] C. Ikuta, Y. Uwate, Y. Nishio, Multi-layer perceptron with pulse glial chain having oscillatory excitation threshold, IEEE Int. Symposium Circuits Systems. IEEE E99.A (3) (2015) 1330–1333.

[15] S. Kang, X. Qian, H. Meng, Multi-distribution deep belief network for speech synthesis, in: International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 8012–8016.

[16] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: International Conference on Machine Learning, 2009, pp. 609–616.

[17] A.Y. Leib, A. Kosovicheva, D. Whitney, Fast ensemble representations for abstract visual impressions, Nature Commun. 7 (2016) 13186.

[18] D. Li, Y. Dong, Deep learning: methods and applications, Found. Trends Inf. Retrieval 7 (3) (2014) 197–387.

[19] P. Liu, S. Han, Z. Meng, Y. Tong, Facial expression recognition via a boosted deep belief network, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1805–1812.

[20] M. Mahrooghy, J.V. Aanstoos, R.A.A Nobrega, K. Hasan, S. Prasad, N.H. Younan, A machine learning framework for detecting landslides on earthen levees using spaceborne SAR imagery, IEEE J. Selected Topics Appl. Earth Observ. Remote Sens. 8 (8) (2017) 3791–3801.

[21] A.R. Mohamed, G.E. Dahl, G. Hinton, Acoustic modeling using deep belief networks, IEEE Trans. Audio Speech Lang. Process 20 (1) (2011) 14–22.

[22] A.R. Mohamed, T.N. Sainath, G. Dahl, B. Ramabhadran, G.E. Hinton, M.A. Picheny, Deep Belief Networks using discriminative features for phone recognition, in: IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2011, pp. 5060–5063.

[23] A. Petropoulos, S.P. Chatzis, S. Xanthopoulos, A hidden Markov model with dependence jumps for predictive modeling of multidimensional time-series, Inf. Sci. (2017), doi:10.1016/j.ins.2017.05.038.

[24] N.H. Phan, X. Wu, D. Dou, Preserving differential privacy in convolutional deep belief networks, Mach. Learn. (2017) 1–24.

[25] J. Qiao, G. Wang, X. Li, W. Li, A self-organizing deep belief network for nonlinear system modeling, Appl. Soft Comput. (2018), doi:10.1016/j.asoc.2018.01.019.

[26] P.P. Roy, Y. Chherawala, M. Cheriet, Deep-belief-network based rescoring approach for handwritten word recognition, in: 14th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2014, pp. 506–511.

[27] H.B. Sailor, H.A. Patil, Novel unsupervised auditory filterbank learning using convolutional RBM for speech recognition, IEEE/ACM Trans. Audio Speech Lang. Process. 24 (12) (2016) 2341–2353.

[28] R. Salakhutdinov, G. Hinton, An efficient learning procedure for deep Boltzmann machines, Neural Comput. 24 (8) (2012) 1967.

[29] A. Sankaran, M. Vatsa, R. Singh, A. Majumdar, Group sparse autoencoder, Image Vis. Comput. (2017), doi:10.1016/j.imavis.2017.01.005.

[30] R. Sarikaya, G.E. Hinton, A. Deoras, Application of deep belief networks for natural language understanding, IEEE/ACM Trans. Audio Speech Lang. Process. 22 (4) (2014) 778–784.

[31] H. Shao, H. Jiang, F. Wang, Y. Wang, Rolling bearing fault diagnosis using adaptive deep belief network with dual-tree complex wavelet packet, ISA Trans. 69 (2017) 187–201.

[32] R. Tanase, M. Datcu, R. Dan, A convolutional deep belief network for polarimetric SAR data feature extraction, Geosci. Remote Sens. Symposium (2016) 7545–7548.

[33] D.N. Tu, T. Tran, D. Phung, S. Venkatesh, Graph-induced restricted Boltzmann machines for document modeling, Inf. Sci. Int. J. 328 (C) (2016) 60–75.

[34] D. Turcsany, A. Bargiela, T. Maul, Local receptive field constrained deep networks, Inf. Sci. 349-350 (C) (2016) 229–247.

[35] C.Y. Wang, J.C. Wang, A. Santoso, C.C. Chiang, C.H. Wu, Sound event recognition using auditory-receptive-field binary pattern and hierarchical-diving deep belief network, IEEE/ACM Trans. Audio Speech Lang. Process. (2017), doi:10.1109/TASLP.2017.2738443.

[36] G. Wang, J. Qiao, X. Li, L. Wang, X. Qian, Improved classification with semi-supervised deep belief network, World Congress Int. Feder. Autom. Control 50 (1) (2017) 4174–4179.

[37] S. Wang, G. Hua, G. Hao, C. Xie, A cycle deep belief network model for multivariate time series classification, Math. Probl. Eng. 2017 (2) (2017) 1–7.

[38] Y. Wang, X. Han, Z. Liu, D. Luo, X. Wu, Modelling inter-task relations to transfer robot skills with three-way RBMs, in: IEEE International Conference on Mechatronics and Automation, 2015, pp. 1276–1282.

[39] R. Zand, K.Y. Camsari, I. Ahmed, S.D. Pyle, C.H. Kim, S. Datta, R.F. DeMara, R-DBN: a resistive deep belief network architecture leveraging the intrinsic behavior of probabilistic devices, 2017. arXiv:1710.00249.

[40] C. Zhang, P. Lim, A.K. Qin, K.C. Tan, Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2017) 2306–2318.

[41] B.H. Zheng, Material procedure quality forecast based on genetic BP neural network, Mod. Phys. Lett. B (2017) 1740080.