An Optimized Dimensionality Reduction Model for High-dimensional Data Based on Restricted Boltzmann Machines

Ke Zhang [*, a, b], Jianhuan Liu [a], Yi Chai [a, b], Kun Qian[a], Jiayi Zhou[a]

[a] Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, College of Automation, Chongqing University, Chongqing 400030, P.R.China

[b] State Key Laboratory of Power Transmission Equipment & System Security and New Technology, College of Automation, Chongqing University, Chongqing 400030, P.R.China

Abstract: In high-dimensional data clustering analysis, a widely used method is to reduce the dimensions of high-dimensional data and do clustering analysis in the relatively low dimensions. Traditional approaches based on multivariate statistical methods, which mainly include Principal Component Analysis (PCA), Self-organized Feature Maps (SOM), Multidimensional scaling (MDS) and Fractal Dimensionality Reduction (FDR), may encounter many deficiencies in solving dimensionality reduction problems. The reason why the described multivariate statistical methods sometime fail to solve dimensionality reduction problems is that before using these methods, specific distribution or enough prior information should be obtained. Unfortunately, high-dimensional data are usually lacking in specific distribution or enough prior information. As a result, it seems unrealistic to deal with dimensionality reduction in many situations with these traditional methods. To solve the problem described above, in this research, an optimized dimensionality reduction model based on Restricted Boltzmann Machines (RBM) is presented. The strength of this model is that data distribution and prior information are not required. In addition, the model was optimized through adjusting the RBM hidden layer structure dynamically. As a result, the optimized model makes the hidden layer units become self-organized. Compared with the fixed-units model, a better result is achieved while fewer amounts of hidden units are used. Meanwhile, this optimized model also is used to classify handwritten digit. Under almost same classification accuracy, hidden layer units are reduced significantly. In conclusion, the model performs well.

Key words: High-dimensional data, clustering analysis, dimensionality reduction, Restricted Boltzmann Machines

1. Introduction

In statistical processing, a lot of information and data, such as stock market trade data, video data of multimedia, aerospace data and bio-character data, are required to be processed immediately both in commercial applications and the scientific research fields. These data are commonly referred to as high- dimensional data.

The features of high-dimensional data can be generally summarized as follows:

(1) Sparsity, Phenomenon of Empty Space and Dimension Effect [1].
(2) The critical problem existing in analyzing high-dimensional data is the Dimension Effect (it is also known as "curse of dimensionality").
(3) High-dimensional data clustering analysis is a popular method to solve the problem caused by the continuous increasing number of data dimensions.

The approach encounters the following three difficulties:

(1) Distance function is difficult to be defined;
(2) Clustering operation can not be performed since clustering centers can not be distinguished clearly;
(3) The complexity of a clustering algorithm increase as the dimensionality of the data increase and this limits the wider application of traditional method.

For another perspective, "curse of dimensionality" is also "blessing of dimensionality". That is, high-dimensional data convey more abundant information to be utilized. Thus, dimensionality reduction as an available method to overcome the curse of dimensionality has attracted extensive attentions. The correlative research is in the ascendant.

During recent years, high-dimensional data clustering analysis has become a more and more popular area in data mining field. The existing methods, which are commonly used in high-dimensional data clustering, can be summarized as Dimensionality Reduction, High-dimensional Data Indexing, High-dimensional Outlier Detection and so on.

Researchers are extremely concerning for mapping relationship between high-dimensional data and its low-dimensional representations in dimensionality reduction methods. Generally speaking, dimensionality reduction mapping methods need to be constructed to obtain the relationship. In many cases, data dimensions should be reduced to a reasonable size. At the same time, the original information should be preserved as much as possible. After dimensionality reduction, all data are sent to the processing system.

As an important branch of data science, dimensionality reduction methods are various. A lot of clustering methods have been proposed based on dimensionality reduction. They mainly include Self-organized Features Maps (SOM) [2-4], Principal Component Analysis (PCA) [5-8], Multidimensional scaling (MDS) [9-11], Fractal Dimensionality Reduction (FDR) [12-14], etc.

Constructing an efficient high-dimensional indexing structure together with improving the performance of querying the similarity of high-dimensional are the research priorities of high-dimensional data indexing technology. As R-Tree and B-Tree, their searching and querying performance will lead to a sharp decline with the increasing of the dimensions. To address this drawback, A-Tree and IQ-Tree have been proposed [1].

The sparse distribution of high-dimensional data makes the high-dimensional outlier monitoring become difficult. Therefore, traditional methods based on the density, distance, depth and deviation are affected [15]. For multiple types of high-dimensional data, the validity of traditional methods, which based on distance to measure the similarities of objects, will decrease a lot. Searching for the adjacent points becomes unstable and difficult.

In this research, the dimensionality reduction of high-dimensional data is treated as the main research content. Actual data generally do not have a special distribution and enough prior information. Therefore, the application of dimensionality reduction based on multivariate statistical analysis is limited. In order to address this problem, Restricted Boltzmann Machine (RBM) is used to fit discrete distribution of high-dimensional data in low-dimensional space.

In response to above problems, a dimensionality reduction model, which has a fixed number of hidden layer units, is constructed. And then, the model is proposed to be improved based on the output strength of hidden layer. This improvement leads to change the number of units in the hidden layer dynamically. Tested by MNIST datasets a better result has been acquired. Meanwhile, the improved model also has been used in handwritten digit classification. Under the same classification accuracy, the improved model significantly reduce the number of hidden layer units and improve efficiency.

About this paper, Part 1 describes the significance and problems of this research. Part 2 describes the related researches currently and mainly focuses on the most popular methods for dimensionality reduction and their advantages and disadvantages. Part 3 describes a dimensionality reduction based on RBM. Part 4 improves the above model presented in Part 3 and proposes a concrete improvement process. The remaining parts design some experiments and display the results analysis and conclusions.

2. Related Researches

Dimensionality reduction has a wide application. A variety of concrete methods and algorithms have been proposed. According to the basic idea of dimensionality reduction, dimensionality reduction algorithms are divided into four types. They are dimensionality reduction based on low-dimensional projection, neural networks, data similarity and fractality (shown in Figure 1).

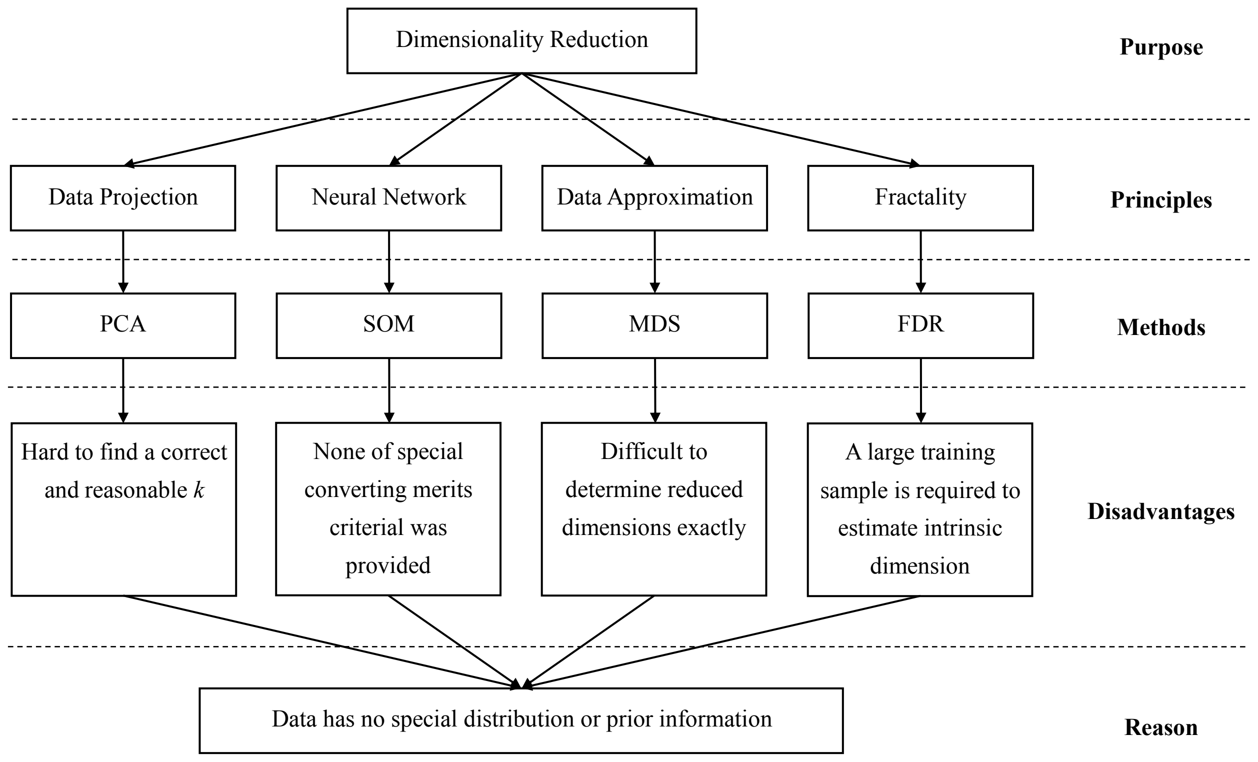

Dimensionality Reduction
Purpose
Data Projection
Neural Network
Data Approximation
Fractality
Principles
PCA
SOM
MDS
FDR
Methods
Hard to find a correct and reasonable k
None of special converting merits criterial was provided
Difficult to determine reduced dimensions exactly
A large training sample is required to estimate intrinsic dimension
Disadvantages
Data has no special distribution or prior information
Reason

Figure 1. Widely used dimension reduction methods and their disadvantages

The purpose of dimensionality reduction is to seek an equivalent low-dimensional representation of high-dimensional data, no matter what kind of dimensionality reduction methods are used. Different way of low-dimension representation will lead to different performance in dimensionality reduction. Therefore, different dimensionality reduction methods have different results.

SOM is an approach based on neural network. Through this method, a low-dimensional mapping figure of high-dimensional data with the premise of retaining the approximate relationship of the data is searched. High-dimensional data clustering processing based on SOM is a typical projective clustering solution. According to this solution, each neuron in the competitive layer must be competed with each other. And the weights of the winners and their neighbor neurons are updated to make them as similar as possible to the input data. After training the neural network, each high-dimensional data is projected onto these neurons according to the matching of these data and weights.

The disadvantage of SOM is that none of specific evaluation criterion has been provided. The criterion should be able to be used to evaluate the converting merits from high-dimension to low-dimension. Furthermore, for high-dimensional data, the neural network training process will be very slow.

PCA is one of the best widespread dimensionality reduction methods. For a data set that contains n m-dimensional data, a $m*m$ covariance matrix is calculated and then the k leading feature vectors of the matrix are computed. The main characteristics of raw data are represented by these k leading feature vectors. Depending on this basis, the raw high-dimensional data can be projected to the direction represented by the k feature vectors. Since the projected data have a relatively low dimension, the conventional clustering algorithm can be used for following clustering.

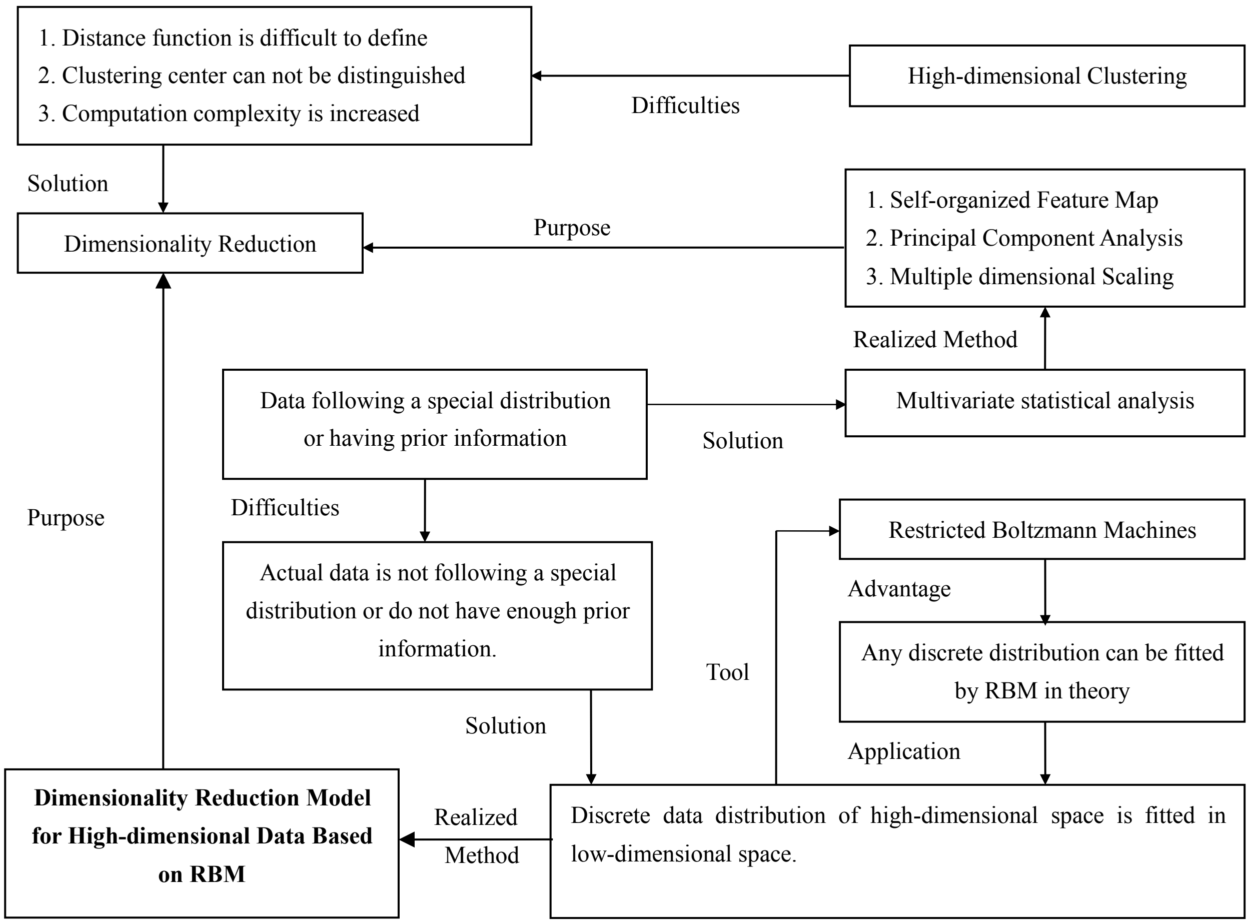

1. Distance function is difficult to define
2. Clustering center can not be distinguished
3. Computation complexity is increased
Difficulties
High-dimensional Clustering
Solution
Dimensionality Reduction
Purpose
1. Self-organized Feature Map
2. Principal Component Analysis
3. Multiple dimensional Scaling
Realized Method
Data following a special distribution or having prior information
Solution
Multivariate statistical analysis
Purpose
Difficulties
Restricted Boltzmann Machines
Actual data is not following a special distribution or do not have enough prior information.
Advantage
Any discrete distribution can be fitted by RBM in theory
Tool
Solution
Application
Dimensionality Reduction Model for High-dimensional Data Based on RBM
Realized Method
Discrete data distribution of high-dimensional space is fitted in low-dimensional space.

Figure 2. reasons why choose RBM for dimensionality reduction

Although several methods to determine the value of k have been provided in PCA, the value of k determined by different methods varies greatly. Thus, it is difficult to find a correct and reasonable k. If k is too small, some important features of the raw data will be lost. On the contrary, a too large k will result in a still high data dimension in spite of retaining most original data information. Accordingly, clustering process will be still hard. Other disadvantages of PCA are space complexity and time complexity. Its space complexity is $o(m^2)$ and its time complexity depending on the number of characteristic values is generally more greater than $o(m^2)$.

MDS is also a popular dimensionality reduction method. In this method, high-dimensional data can be mapped to low-dimensional space. The differences between data points can be retained in the mapping process. The points closed to each other in the original data remains close to each other, and those points far away from each other in the original data still remains far away. The basic starting point of this algorithm has the feature of describing the similarity between data points. The goal of dimension reduction is to obtain the corresponding high-dimensional data features. These features can be used to simplify the analysis, get effective features and make data visible through the analysis of low-dimensional data. Therefore, as long as remaining the difference between the data sufficiently, an effective low-dimensional representation can be obtained.

One disadvantage of MDS is that no good principle has been provided to determine how many dimensions the data should be reduced to exactly. Furthermore, time complexity of most of such methods is $o(n^2)$, where n is the size of the data set.

FDR is get attention as a class of dimensionality reduction methods. The intrinsic dimensionality of data can be estimated accurately using fractality, so as to provide guidance for further dimensionality reduction. Compared with other methods to estimate the intrinsic dimension, non-integer values of intrinsic dimension can be obtained, known as fractal dimension. There are many descriptions about the definition of fractal dimension. Box-counting dimension and correlation dimension are widely used. A lot of different methods are resulted based on these estimations of corresponding dimensions.

In summary, for the above processing methods, there are many difficulties that need to be solved. For example, with the increasing of dimension, the calculated amount of data will increase rapidly and the number of samples in space will decrease. Because of these difficulties, some statistical progressivity is difficult to be achieved. While dealing with actual data using traditional multivariate statistical analysis, some difficulties will be encountered. These difficulties include that data do not follow a normal distribution or do not have much prior information. To address this problem, using non-parametric methods is effective. A model requires to be found and in this model, although a particular data distribution and prior information are not existed or sufficient, discrete data distribution of high-dimensional space can be fitted in the low-dimensional space.

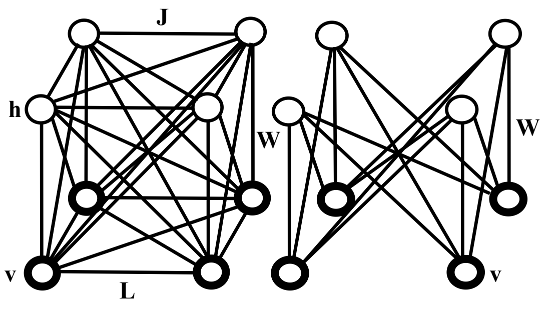


Figure 3. Left: A general Boltzmann machine. The top layer represents a vector of stochastic binary "hidden" features and the bottom layer represents a vector of stochastic binary "visible" variables. Right: A restricted Boltzmann machine with no hidden-to-hidden and no visible-to-visible connections.

Boltzmann Machine (BM) is a random neural networks rooted in statistical mechanics. The network status is determined by the value of statistical probability. The BM's disadvantages mainly include the case that long training time is necessary and it is difficult to get a random sample following BM distribution. All these disadvantages can be overcome by Restricted Boltzmann Machine. The activation condition of each hidden unit is independent when given the state of the visible layer units. Conversely, the activation condition of each visible unit is independent when given the state of hidden layer units [16]. Based on the above descriptions, the random samples through Gibbs sampling can be obtained [17]. In theory, as long as there are enough hidden units, RBM can be used to fit any discrete distributions [18] (shown in Figure 2).

In this research, in order to construct a model of dimensionality reduction of high-dimensional data, a deep auto-encoder has been trained based on RBM. To achieve required accuracy, how many units the traditional RBM hidden layer should be set is uncertain. A new way of setting hidden units number is presented. In this way, a mechanism is applied to achieve the purpose of dynamic changing RBM hidden layers. Compared with a fixed number of hidden layers, a better performance is obtained by testing the MNIST database.

3. Dimensionality Reduction Model

A Boltzmann machine is a network of symmetrically coupled stochastic binary units [19]. It contains a set of visible units $\mathbf{v} \in \{0,1\}^D$, and a set of hidden units $\mathbf{h} \in \{0,1\}^P$ (showed in Figure 3). The energy of the state $\{\mathbf{v},\mathbf{h}\}$ is defined as:

$$\mathrm{E}(\mathbf{v},\mathbf{h};\theta) = -\frac{1}{2}\mathbf{v}^{\mathrm{T}}\mathbf{L}\mathbf{v} - \frac{1}{2}\mathbf{h}^{\mathrm{T}}\mathbf{J}\mathbf{h} - \mathbf{v}^{\mathrm{T}}\mathbf{W}\mathbf{h} \tag{1}$$

Where $\theta = \{\mathbf{W}, \mathbf{L}, \mathbf{J}\}$ are the model parameters: **W**, **L**, **J** represent visible-to-hidden, visible-to-visible, and hidden-to-hidden symmetric interaction terms respectively. The probability that the model assigns to a visible vector **v** is:

$$p(\mathbf{v};\theta) = \frac{p^*(\mathbf{v};\theta)}{Z(\theta)} = \frac{1}{Z(\theta)}\sum_h \exp(-E(\mathbf{v},\mathbf{h};\theta)) \tag{2}$$

$$Z(\theta) = \sum_{\mathbf{v}}\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h};\theta)) \tag{3}$$

where p^* denotes un-normalized probability, and $Z(\theta)$ is the partition function. The conditional distributions over hidden and visible units are given by:

$$p(h_j = 1 \mid \mathbf{v}, \mathbf{h}_{-j}) = \sigma\left(\sum_{i=1}^{D} W_{ij} v_i + \sum_{m=1\setminus j}^{P} J_{jm} h_j\right) \tag{4}$$

$$p(v_i = 1 \mid \mathbf{h}, \mathbf{v}_{-i}) = \sigma\left(\sum_{j=1}^{P} W_{ij} h_j + \sum_{k=1\setminus i}^{D} L_{ik} v_j\right) \tag{5}$$

where $\sigma(x)=1/(1+\exp(-x))$ is the logistic function. The parameter updates are needed to perform gradient ascent in the log-likelihood can be obtained from Eq. 2:

$$\Delta\mathbf{W} = \alpha\left(\mathrm{E}_{P_{\text{data}}}\left[\mathbf{v}\mathbf{h}^{\mathrm{T}}\right] - \mathrm{E}_{P_{\text{recon}}}\left[\mathbf{v}\mathbf{h}^{\mathrm{T}}\right]\right) \tag{6}$$

$$\Delta\mathbf{L} = \alpha\left(\mathrm{E}_{P_{\text{data}}}\left[\mathbf{v}\mathbf{v}^{\mathrm{T}}\right] - \mathrm{E}_{P_{\text{recon}}}\left[\mathbf{v}\mathbf{v}^{\mathrm{T}}\right]\right) \tag{7}$$

$$\Delta \mathbf{J} = \alpha\left(\mathrm{E}_{P_{\text{data}}}\left[\mathbf{h}\mathbf{h}^{\mathrm{T}}\right] - \mathrm{E}_{P_{\text{recon}}}\left[\mathbf{h}\mathbf{h}^{\mathrm{T}}\right]\right) \tag{8}$$

where α is a learning rate, $\mathrm{E}_{P\text{data}}[\cdot]$ denotes an expectation with respect to the completed data distribution

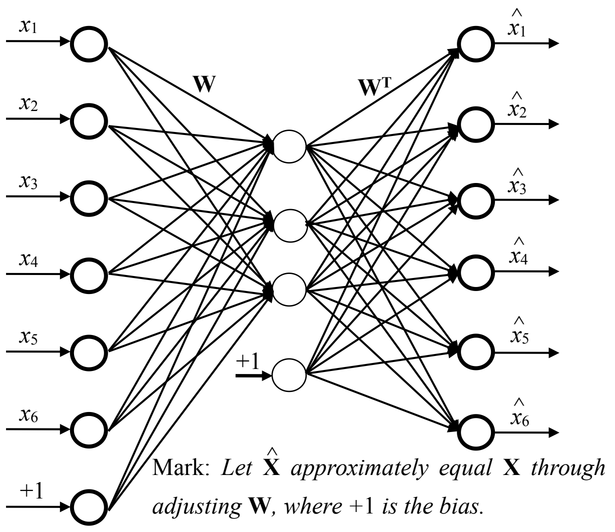
$$P_{\text{data}}(\mathbf{h}, \mathbf{v}; \theta) = p(\mathbf{h} \mid \mathbf{v}; \theta) P_{\text{data}}(\mathbf{v}) \tag{9}$$

with

$$P_{\text{data}}(\mathbf{v}) = \frac{1}{N}\sum_{n} \delta(\mathbf{v} - \mathbf{v}_n) \tag{10}$$

representing the empirical distribution, and $\mathrm{E}_{P_{recon}}[\cdot]$ is an expectation with respect to the distribution defined by the model (see Eq. 2).

Setting both **J**=0 and **L**=0 recovers the well-known restricted Boltzmann machine (RBM) model [20] (shown in Figure 3, right panel). In contrast to general BM's, inference in RBM's is exact. Although exact maximum likelihood learning in RBM's is still intractable, learning can be carried out efficiently by using Contrastive Divergence (CD) [21]. It was further observed that for Contrastive Divergence to perform well, it is important to obtain exact samples from the conditional distribution $p(\mathbf{h}|\mathbf{v};\theta)$[22], which is intractable when learning full Boltzmann machines.

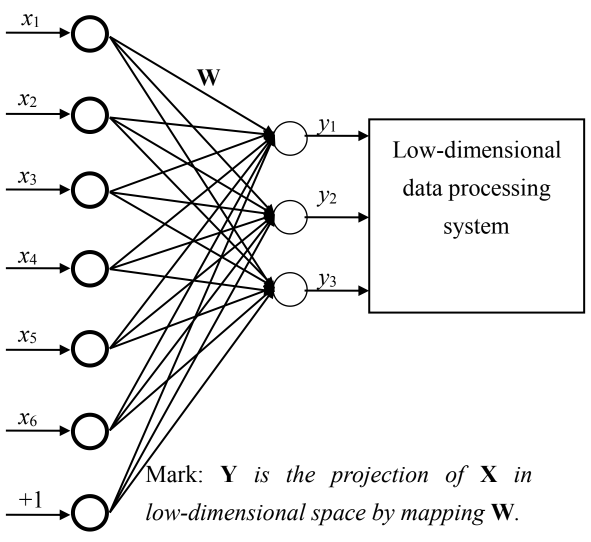


x_1
x_2
x_3
x_4
x_5
x_6
+1
W
$\mathbf{W}^{\mathrm{T}}$
+1
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_4$
$\hat{x}_5$
$\hat{x}_6$
Mark: *Let* $\hat{\mathbf{X}}$ *approximately equal* **X** *through adjusting* **W**, *where* +1 *is the bias.*

Figure 4. Structure of Auto-encoder

In 2006, a greedy layer-by-layer training algorithm was proposed by Hinton *et al*. In this algorithm, a layer of RBM is trained each time, and then the output of this RBM is regarded as the input of the next RBM [23]. Since then, RBM began to be widely used, especially in deep structures.

Auto-encoder is a neural network with multiple hidden neurons (shown in Figure 4). The data-encoding and data-decoding can be completed through adjusting the network structure. Data information is stored in the network weights after training and then the weights will be used to decode the input data. Compared with RBM, many similarities between data decoding of auto-encoder and reconstructing process of RBM can be found obviously. So they are equivalent in essence. After the output layer is removed in Figure 4, the value of the hidden layer is the expected feature (shown in Figure 5). Apparently, when the reconstructed data and the original data are approximately equal, encoding process is a feature transforming process from visible layer to hidden layer. In this process the dimensionality reduction of high-dimensional data is achieved.



x_1
x_2
x_3
x_4
x_5
x_6
+1
W
y_1
y_2
y_3
Low-dimensional data processing system
Mark: **Y** *is the projection of* **X** *in low-dimensional space by mapping* **W**.

Figure 5.Process of Features Selecting

To construct a deep auto-encoder, four layers of RBMs should be trained. The algorithm process can be divided into three steps which include pre-training, unrolling and fine-tuning (shown in Figure 6).

Pre-training stage consists of learning a stack of restricted Boltzmann machines, each having only one layer of feature detectors. After learning one layer of feature detectors, their activities are treated-when they are being driven by the data-as data for learning a second layer of features. Feature detectors of the first layer then become the visible units for learning the next RBM. This layer-by-layer learning can be repeated as many times as desired.

After pre-training multiple layers of feature detectors, the model is a folded auto-encoder. It can be unfolded to produce encoder and decoder networks which initially have the same weights.

In the global fine-tuning stage, stochastic activities are replaced by deterministic, real-valued probabilities. And back-propagation through the whole auto-encoder to fine-tune the weights is used for optimal reconstruction.

For this model, the visible units of every RBM had real-valued activities. These activities were in the range [0, 1] for logistic units. While training higher level RBM, the visible units were set to the activation probabilities of the hidden units in the previous RBM, but the hidden units of every RBM except the top one had stochastic binary values. The hidden units of the top RBM had stochastic real-valued states drawn from a unit variance Gaussian whose mean was determined by the input from that RBM's logistic visible units [24].

4. Improved Dimensionality Reduction Model

It is left to be solved that how many units should be selected in the hidden layers of RBM. Too few units will not satisfy the required accuracy, while too many units can satisfy high accuracy but sacrificing the training time. It is of great importance to select the appropriate number of hidden layer units to satisfy both accuracy and efficiency.

Intuitions derived from discriminative machine learning are a bad guide for determining a sensible number of hidden units. In discriminative learning, the amount of constraints that a training case imposes on the parameters is equal to the number of bits that it takes to specify the label. Labels usually contain very few bits of information, so using more parameters than training cases will typically cause severe over-fitting.

When learning generative models of high-dimensional data, however, it is the number of bits that it takes to specify a data vector that determines how many constraints each training case imposes on the parameters of the model. This can be several orders of magnitude greater than numbers of bits required to specify a label. So it may be quite reasonable to fit a million parameters to 10000 training images if each image contains 1000 pixels. This would allow 1000 globally connected hidden units. If the hidden units are locally connected or if they use weight-sharing, many more can be used [25].

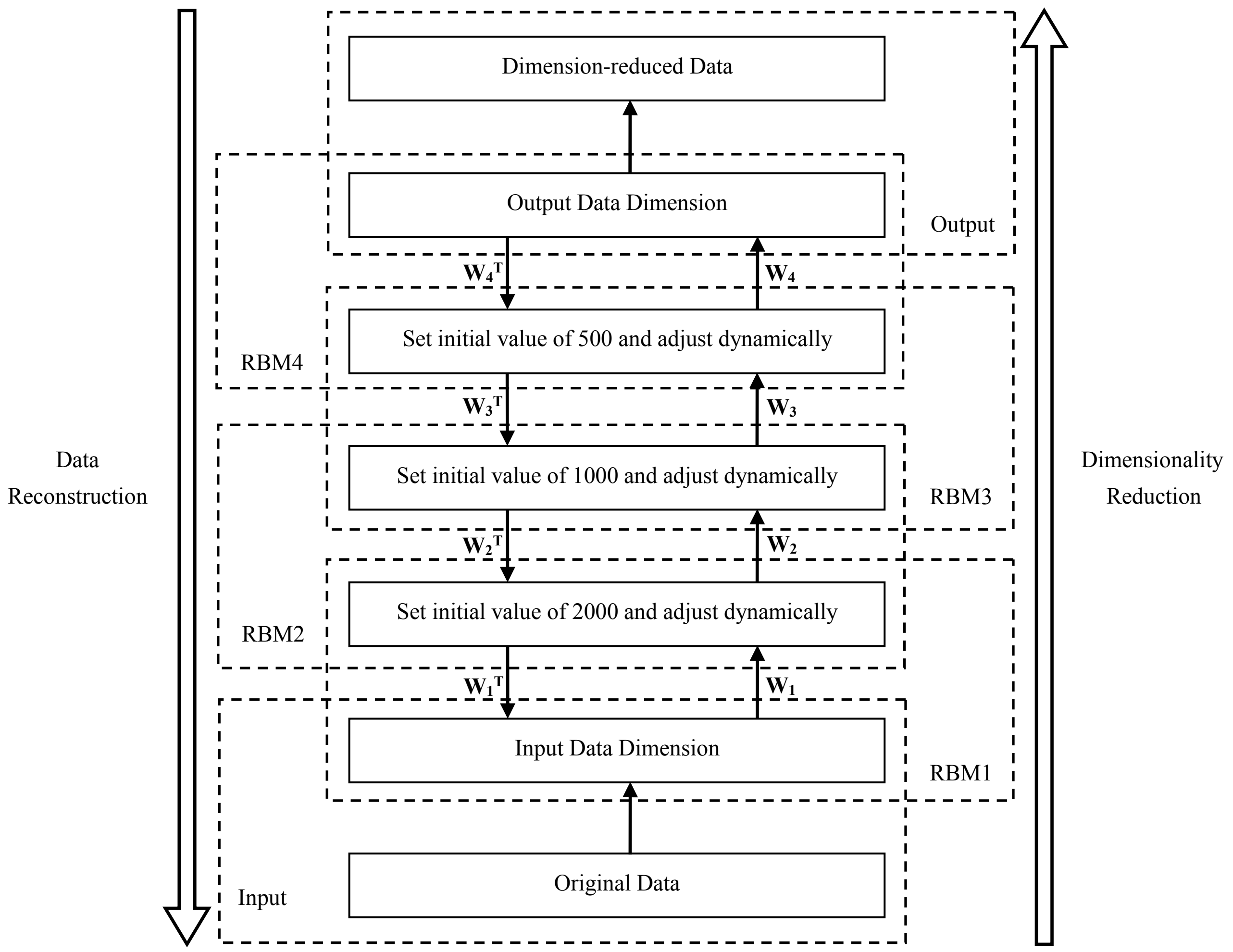


Figure 6. Dimensionality Reduction Model based on RBM

Assuming that the main issue is over-fitting rather than the amount of computation at training or test time, estimate how many bits it would take to describe each data-vector if a good model was used. Then multiply that estimate by the number of training cases and use a number of parameters that is about an order of magnitude smaller. If a sparsity target that is very small is required, more hidden units can be used. If the training cases are highly redundant, as they typically will be for very big training sets, fewer parameters can be used [25].

In this research, the dimensionality reduction model mentioned above has been improved by adjusting the hidden structure dynamically. This adjustment is based on the Output Information Intensity. With Output Information Intensity, compute the active degree of hidden units, then split the hidden units with strong active degree, but the weak ones should be deleted. With all the work done, modify the structure parameters, making sure that the hidden neural units are adjusted dynamically. It has been proved that the approach proposed here performs well.

To achieve RBM hidden layer structure adjustment, the active degree using the output of the hidden layer will be calculated based on:

$$Af_i(x) = \frac{\theta_i}{\sum_{k=1}^{K} \theta_k} \tag{11}$$

where *i=1, 2,…,K*, Af_i is the active degree of unit *i*, *K* is the total number of hidden units and θ_i is the output of hidden units. Unit will be split when Af_i is greater than the threshold Af_0. According to the number of new units, the connection weights are set based on:

$$w_{i,l} = r_l \frac{w_i\theta_i(x) - e}{\theta_{i,l}(x)} \quad \sum_{l=1}^{N_{new}} r_l = 1 \tag{12}$$

where r_l is the assigned parameter of new units, θ_i is the output of hidden units before divided, $\theta_{i,\,l}(x)$ is the output of new units and *e* is the error before divided. When Af_i is less than the threshold Af_1, the unit, its connection and bias will all be deleted directly. Convergence analysis is given below.

$$e'_{n+N_{new}-1}(t) = \sum_{k=1}^{n+N_{new}-1} \omega_k\theta_k(x(t)) - y_d(t) =$$

$$\sum_{k=1}^{n} \omega_k\theta_k(x(t)) - \omega_i\theta_i(x(t)) + \varphi(x(t)) - y_d(t) \tag{13}$$

where

$$\varphi(x(t)) = \sum_{j=1}^{N_{new}} \omega_{i,j}\theta_j(x(t)) \tag{14}$$

Based on the given parameters adjustment rules Eq. 12,

$$\sum_{l=1}^{N_{new}} \omega_{i,l}\theta_l(x(t)) - \omega_i\theta_i(x(t)) =$$

$$\sum_{l=1}^{N_{new}} r_l \frac{\omega_i\theta_i(x) - e_n(t)}{\theta_l(x)}\theta_l(x(t)) - \omega_i\theta_i(x(t)) = -e_n(t) \tag{15}$$

$$e'_{n+N_{new}-1}(t) = \sum_{k=1}^{n} \omega_k\theta_k(x(t)) - \omega_i\theta_i(x(t)) +$$

$$\varphi(x(t)) - y_d(t) = \sum_{k=1}^{n} \omega_k\theta_k(x(t)) - y_d(t) -$$

$$e_n(t) = e_n(t) - e_n(t) = 0 \tag{16}$$

The above analysis shows that, the output error will converge after the restructuring. From another perspective , the division of neurons can improve learning efficiency and network performance.

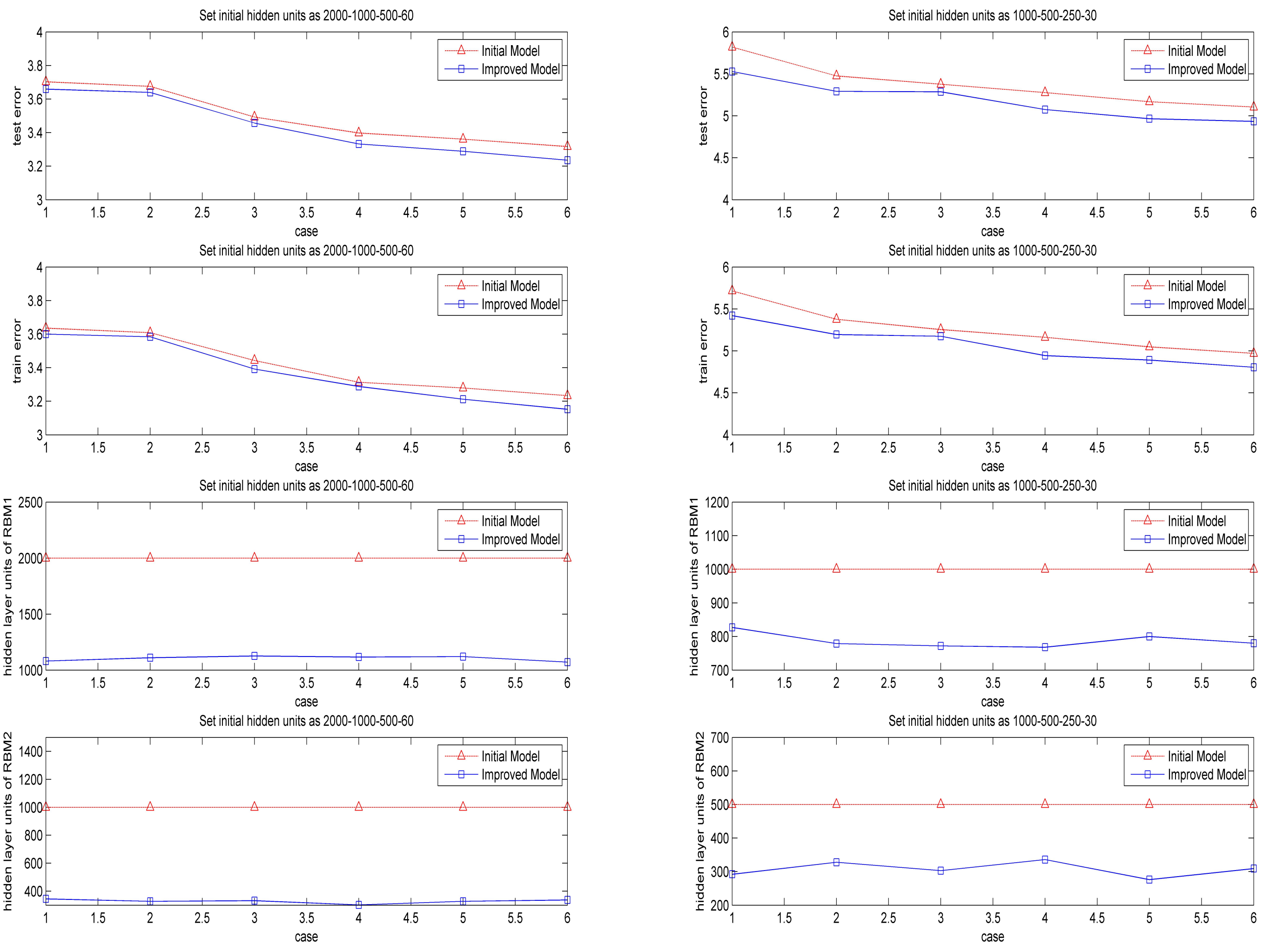


Figure 7. Parameter and Error Variations of Initial RBM Model and Improved RBM Model

Table 1. Comparison of Initial RBM Model and Improved RBM Model

Initial Number	Index	RBM1			RBM2			RBM3			RBM4			Reconstruct		
		Itera	num	error	Itera	num	error	Itera	num	error	Itera	num	error	Itera	Train error	Test error
2000 | 1000 | 500 | 60	1	8	2000	0.6966	15	1000	0.7672	4	500	0.5396	26	60	0.9794	25	3.6356	3.7025
	2	11	**1081**	0.6859	20	**345**	0.7562	6	500	0.5484	31	60	0.9842	25	**3.6002**	**3.6592**
	1	8	2000	0.6987	14	1000	0.7761	4	500	0.5363	27	60	0.9755	30	3.6089	3.6755
	2	9	**1110**	0.6885	24	**328**	0.7728	6	500	0.5490	25	60	0.9783	30	**3.5840**	**3.6397**
	1	8	2000	0.6993	16	1000	0.7671	4	500	0.5422	31	60	0.9805	35	3.4435	3.4928
	2	8	**1126**	0.6892	27	**332**	0.7772	6	500	0.5266	25	60	0.9887	35	**3.3920**	**3.4571**
	1	10	2000	0.6948	15	1000	0.7608	4	500	0.5293	28	60	0.9712	40	3.3130	3.3978
	2	8	**1117**	0.6987	37	**302**	0.7768	5	500	0.5485	22	60	0.9811	40	**3.2883**	**3.3325**
	1	8	2000	0.6975	14	1000	0.7781	4	500	0.5418	29	60	0.9790	45	3.2794	3.3611
	2	10	**1121**	0.6779	24	**328**	0.7718	5	500	0.5191	25	60	0.9803	45	**3.2121**	**3.2889**
	1	7	2000	0.6985	17	1000	0.7644	5	500	0.5214	33	60	0.9796	50	3.2337	3.3169
	2	8	**1071**	0.6988	22	**337**	0.7798	6	500	0.5452	24	60	0.9884	50	**3.1524**	**3.2357**
1000 | 500 | 250 | 30	1	8	1000	0.6993	24	500	0.7591	4	250	0.5195	14	30	0.9709	25	5.7160	5.8187
	2	11	**827**	0.6991	27	**292**	0.7776	11	250	0.5315	16	30	0.9866	25	**5.4209**	**5.5273**
	1	8	1000	0.6890	17	500	0.7771	5	250	0.5383	13	30	0.9817	30	5.3772	5.4756
	2	10	**779**	0.6933	23	**328**	0.7464	11	250	0.5497	16	30	0.9882	30	**5.1947**	**5.2919**
	1	8	1000	0.6987	17	500	0.7756	6	250	0.5394	17	30	0.9741	35	5.2544	5.3763
	2	10	**772**	0.6885	21	**303**	0.7673	17	255	0.5324	18	30	0.9699	35	**5.1753**	**5.2861**
	1	7	1000	0.6989	14	500	0.7787	5	250	0.5442	14	30	0.9874	40	5.1621	5.2766
	2	8	**768**	0.6965	21	**336**	0.7760	16	250	0.5325	17	30	0.9805	40	**4.9439**	**5.0754**
	1	8	1000	0.6915	14	500	0.7762	7	250	0.5373	16	30	0.9845	45	5.0484	5.1689
	2	10	**800**	0.6888	23	**276**	0.7785	9	250	0.5416	14	30	0.9724	45	**4.8906**	**4.9658**
	1	9	1000	0.6949	16	500	0.9601	5	250	0.5351	18	30	0.9609	50	4.9704	5.1048
	2	11	**780**	0.6985	24	**309**	0.7687	11	250	0.5455	22	30	0.9801	50	**4.8050**	**4.9334**
500 | 250 | 125 | 15	1	11	500	0.6997	23	250	0.7402	52	125	0.5421	11	15	0.9557	25	10.0101	10.0034
	2	10	494	0.6918	21	247	0.7697	42	125	0.5492	8	15	0.9816	25	10.3481	10.3341
	1	12	500	0.6974	16	250	0.7626	66	125	0.5491	23	15	0.9802	30	9.4544	9.5084
	2	14	494	0.6865	27	244	0.7709	37	125	0.5676	12	15	0.9890	30	10.0117	9.9949
	1	11	500	0.6980	14	250	0.7708	81	125	0.5431	12	15	0.9897	35	9.1349	9.1788
	2	12	496	0.6958	19	248	0.7610	55	125	0.5382	9	15	0.9647	35	9.6837	9.7109
	1	10	500	0.6959	16	250	0.7477	47	125	0.5500	14	15	0.9733	40	9.3492	9.4379
	2	13	492	0.6995	22	247	0.7550	50	125	0.5486	9	15	0.9871	40	9.6079	9.6536
	1	10	500	0.6964	18	250	0.7773	129	125	0.5455	18	15	0.9834	45	8.8568	8.9085
	2	15	492	0.6940	22	249	0.7681	86	125	0.5386	12	15	0.9770	45	9.3134	9.3556
	1	11	500	0.6879	15	250	0.7578	66	125	0.5485	17	15	0.9869	50	8.7949	8.8335
	2	12	481	0.6858	20	248	0.7714	59	125	0.5392	12	15	0.9685	50	9.0878	9.1645

For structure adjustment, the newly added parameters are generally set to any initial value. This parameters setting method may easily lead to fluctuations in network errors. The presented method can eliminate the error fluctuations caused by structure self-organization, and speed up the convergence of error.

5. Experiment Results

The MNIST datasets are used in our experiment. The MNIST digit dataset contains 60000 training images and 10000 test images of ten handwritten digits, with 28*28 pixels. To speed up learning, the datasets are subdivided into several mini-batches, each containing 100 cases. And the weights are updated after each mini-batch.

In first experiment, the model tests were carried out for several times and the number of hidden layer units were set to 2000-1000-500-60, 1000-500-250-30 and 500-25-125-15. In Table 1, the part of *Index*=1 shows the testing results under a required accuracy after multiple iterations. In this table, for the first experiment, *num* refers to the amount of hidden layer units, *err* using units of 1.0e+03 refers to error of each iteration, *test error* is reconstruction error of test data, *Itera* is the number of iterations and *train error* is reconstruction error of train data.

For second experiment, the model tests also were carried out many times. The initial number of hidden layer units were set to 2000-1000-500-60, 1000-500-250-30 and 500-250-125-15. As the same situation in Table 1, the part of *Index*=2 shows the testing results under a required accuracy after multiple iterations. In this table, for the second experiment, *num* refers to the amount of remained hidden layer units after several iterations, *err* using units of 1.0e+03 refers to error of each iteration, *test error* is reconstruction error of test data, *Itera* is the number of iterations and *train error* is reconstruction error of train data.

Each of RBM's reverse reconstructing errors is set as 0.7, 0.78, 0.55 and 0.99. The process of training current RBM will be stopped when errors meet the expected accuracy. Then the number of iterations and training errors are recorded. For the stage of reconstruction, the iterations are set as 25, 30, 35, 40, 45 and 50. Finally, the training error and testing error of reconstruction process are recorded (shown in Table 1).

As clearly shown in Table 1, more hidden layer unit results in higher reconstruction accuracy. However, the training time will increase, which means the efficiency will reduce with the number of hidden layer units increasing. In real world application, accuracy and efficiency should be taken into account at the same time.

Overall speaking, the improved model does not perform an obviously good result in the case of a smaller number of hidden layer units. However, the improved model can perform well and better results are obtained obviously when the number of hidden layer units is larger. The number of hidden layer units of the improved model is significantly less than the original model. Besides, the final reconstruction error of the improved model is also significantly less than the original model (shown in Figure 7).

Based on the above improved dimensionality reduction model, handwritten digit classification is done. About the classification experiment, a three-layer auto-encoder has been trained and the initial number of hidden layer units is respectively set as 2000-1000-500, 1000-500-250 and 500-250-125. In the initial dimensionality reduction model, the number of the hidden layer units is kept unchanged and in the improved dimensionality reduction model, the hidden layer units number is adjusted through above algorithm presented in Part 4. In this experiment, the model corresponding to the parameters is run five times and the number of classification iteration is set as 20. The experiment results have been shown in Table 2, where includes the classification accuracy and final hidden layer units amount.

Table 2. Classification Results on Handwritten Digit

Model		Initial Model			Improved Model		
Initial Number	Index	Final Number	Train_acc	Test_acc	Final Number	Train_acc	Test_acc
2000-1000-500	1	2000-1000-500	99.963	98.470	**1138-300-500**	99.968	98.490
	2		99.965	98.470	**1168-296-418**	99.975	98.460
	3		99.967	98.460	**1150-287-500**	99.965	98.440
	4		99.965	98.440	**1074-328-486**	99.970	98.380
	5		99.963	98.430	**1128-330-500**	99.965	98.400
1000-500-250	1	1000-500-250	99.975	98.480	**803-313-250**	99.958	98.550
	2		99.970	98.450	**776-342-168**	99.970	98.450
	3		99.966	98.460	**761-363-201**	99.976	98.470
	4		99.968	98.490	**789-367-250**	99.961	98.470
	5		99.970	98.420	**798-328-250**	99.960	98.490
500-250-125	1	500-250-125	99.965	98.470	**488-250-125**	99.963	98.500
	2		99.967	98.530	**492-249-125**	99.973	98.450
	3		99.958	98.470	**493-248-125**	99.958	98.430
	4		99.963	98.490	**492-247-125**	99.968	98.450
	5		99.965	98.480	**496-248-125**	99.945	98.430

As clearly shown in Table 2, the improved model does not perform an obviously good result in the case of a smaller number of hidden layer units. However, the improved model can perform well and better results are obtained obviously when the number of hidden layer units is larger. In each group, the classification accuracy is almost the same, but the number of hidden layer units are reduced significantly and the efficiency is dramatically improved.

6. Conclusions

Lacking special distribution and prior information of actual data is a main problem of dimensionality reduction based on multivariate statistical analysis. To solve this problem, an optimized model for dimensionality reduction is presented in this research. The optimized method is performed by adjusting the number of hidden layer units dynamically. Based on the discussion above, some conclusions are drawn:

(1) The dimensionality reduction model based on RBM performs well because of the advantage of RBM;

(2) Hidden layer units self-organization of the improved model has been achieved through calculating the output information intensity of hidden layer;

(3) Compared with the fixed number model, a better result is achieved while fewer amount of units are used;

(4) Used in handwritten digit classification, the improved model can reach almost the same classification accuracy, meanwhile significantly reduce the hidden layer units;

(5) Although the application effect of improved model is better, it is a method relied on initial number of hidden layer units. Thus, the research needs to be taking a step forward.

In summary, the improved model can be used to reduce dimensionality efficiently. The discrete distribution of high-dimensional data can be fitted in the low-dimensional space.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Grant No.61203084 and 61374135) and the National Natural Science Foundation of Chongqing China (cstc2011jjA40013).

Conflict of Interests

The authors declare that there is not any commercial or associative conflict of interests regarding the publication of this paper.

References

[1] Parsons, L., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, *6*(1), 90-105.

[2] Salas, J. P., Aguera Perez, A., De la Rosa, J. J. G., & Ramiro, J. G. (2010, July). Clustering using SOFM and genetic algorithm. In *Neural Networks (IJCNN), The 2010 International Joint Conference on* (pp. 1-6). IEEE.

[3] Huang, W., & Wu, Q. J. (2010, March). Human action recognition based on self organizing map. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on* (pp. 2130-2133). IEEE.

[4] Gutkin, R., Green, C. J., Vangrattanachai, S., Pinho, S. T., Robinson, P., & Curtis, P. T. (2011). On acoustic emission for failure investigation in CFRP: Pattern recognition and peak frequency analyses. *Mechanical Systems and Signal Processing*, *25*(4), 1393-1407.

[5] Jolliffe, I. (2005). *Principal component analysis*. John Wiley & Sons, Ltd.

[6] Zhou D.Z., Jiang W.B., Li M.Q. Efficient clustering algorithm for multivariate time series[J]. *Computer Engineering and Application*, 2010, 46(1): 137-139.

[7] Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., & Ma, Y. (2013). Robust recovery of subspace structures by low-rank representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *35*(1), 171-184.

[8] Croux, C., Filzmoser, P., & Fritz, H. (2013). Robust sparse principal component analysis. *Technometrics*, *55*(2), 202-214.

[9] Griffiths, T. L., & Kalish, M. L. (2002). A multidimensional scaling approach to mental multiplication. *Memory & cognition*, *30*(1), 97-106.

[10] Young, F. W. (2013). *Multidimensional scaling: History, theory, and applications*. Psychology Press.

[11] Robinson, J. P., & Gershuny, J. (2013). Visualizing multinational daily life via multidimensional scaling (MDS). *electronic International Journal of TTime Use ResearchT*, 76.

[12] Camastra, F., & Vinciarelli, A. (2002). Estimating the intrinsic dimension of data with a fractal-based method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*(10), 1404-1407.

[13] Camastra, F. (2003). Data dimensionality estimation methods: a survey. *Pattern recognition*, *36*(12), 2945-2954.

[14] Mo, D., & Huang, S. H. (2012). Fractal-based intrinsic dimension estimation and its application in dimensionality reduction. *Knowledge and Data Engineering, IEEE Transactions on*, *24*(1), 59-71.

[15] He, Z., Deng, S., Xu, X., & Huang, J. Z. (2006). A fast greedy algorithm for outlier mining. In *Advances in Knowledge Discovery and Data Mining* (pp. 567-576). Springer Berlin Heidelberg.

[16] Freund, Y., & Haussler, D. (1994). *Unsupervised learning of distributions of binary vectors using two layer networks*. Computer Research Laboratory University of California, Santa Cruz.

[17] Le Roux, N., & Bengio, Y. (2008). Representational power of restricted Boltzmann machines and deep belief networks. *Neural Computation*, *20*(6),

1631-1649.

[18] Liu, J. S. (2008). *Monte Carlo strategies in scientific computing*. Springer.

[19] Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. *Cambridge, MA: MIT Press*, *1*, 282-317.

[20] Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory.

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

[22] Welling, M., & Hinton, G. E. (2002). A new learning algorithm for mean field Boltzmann machines. In *Artificial Neural Networks—ICANN 2002* (pp. 351-357). Springer Berlin Heidelberg.

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

[24] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504-507.

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