Fuzzy Systems and Data Mining VII
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA210211

Feature Back-Tracking with Sparse Deep Belief Networks

Chen QIAO[a], Jiajia LI[a], Xuewu ZHANG[b], Cheng ZHANG[b], Wenfeng JING[a,1] and Danglin YANG[c]
[a] *School of Mathematics and Statistics, Xi'an Jiaotong University, China*
[b] *China Railway First Survey and Design Institute Group Co., Ltd, China*
[c] *Suzhou Hanlin Information Technology Development Co., Ltd, China*

Abstract. To find a way of the interpretability of deep learning, in this paper, a features back-tracking (FBT) approach based on a sparse deep learning architecture is proposed. Firstly, for a deep belief network (DBN), both the Kullback-Leibler divergence of the hidden neurons and the L_1 norm penalty on the connection weights are introduced. Thus, the sparse response mechanism as well as the sparse connection of the brain neurons can be simulated directly. That means the DBN can learn a sparse framework and an effective sparse data representation. On this basis, the feature back-tracking technique is put forward. For both the single nucleotide polymorphisms (SNPs) data and MNIST data, FBT has quite well performance on searching for the risk loci on the genes as well as the important sites of the digit data. It reveals that the proposed FBT method can pick out the essential features by deep learning architecture with quite high classification accuracy and data storage ability. Utilizing the sparse layer-wise feature learning to achieve key features from the original data, is an effective attempt to explore the profound mechanism of human brain and interpretability of deep learning.

Keywords. Features back-tracking, Deep belief networks, Sparse learning, Markers selection

1. Introduction

With the quickly growing demand for revealing the intrinsic nature of things under the complex surface phenomena, and with the development of technology methods, high-dimensional data emerge magically every day. To fully explore valuable information contained in such kind of data, models with strong expression ability are needed. As an effective learning approach, deep learning can learn the layer-wise nonlinear expression or features by greedy layer-by-layer training, thus to find out the complex information of the data. In the recent decade, deep learning research has been getting a flourishing development, and setting off a significant booming on various artificial intelligence domains. Its applications have also been extended to more application fields, like audio recognition, social network, automatic control, bioinformatics, etc.

However, for deep learning, there still exist many intrinsic theoretical issues worth to be further clarified. Currently, most of the deep learning methods are used for layer-

[1] Corresponding Author, Wenfeng Jing, School of Mathematics and Statistics, Xi'an Jiaotong University, China; E-mail: wfjing@mail.xjtu.edu.cn.

wise feature extraction and for classification purpose, and one typical issue is the interpretability of the deep learning framework. For interpretability, one of the most commonly used method is feature selection, which can effectively select important features from the original data containing a large number of features and eliminate irrelevant features [1]. With feature selection, the mechanism of recognizing different things by human brains can be explored, and the essential features which help to distinguish different data can be identified. With the direct interpretability of data, feature selection methods are widely applied in text classification, data mining, bioinformatics, computer vision, information retrieval, time series prediction, and so on.

It should be noticed that most of the existing feature selection methods are based on shallow models. They are only performed on the original data directly. Searching for an approach to obtain the essential features of the data from the deep expression with layered structure, will be great helpful to obtain the essential features of the original data in a global and layer-wised way, and what is the most important, to obtain the interpretability of the deep learning framework.

On the other hand, studies of the brain's nervous system have shown that such system employs a highly sparse response mechanism. For each neuron, it has a topology that only some of the other neurons are connected to this neuron. The sparseness of this connection is the sparse structure of the network or sparse topology. It has been shown that there exist connection sparseness properties in the brain's organizational structure [2-4]. The connection sparseness guarantees the high promotion ability of the human neural network system. Sparse connections are more able to provide high-quality storage capabilities and makes the deep network to have better generalization capabilities [2].

How to better simulate the sparseness of the human neuron system and improve the performance of the deep learning algorithm is a hot topic. At present, there are three main approaches to get the topology sparseness of a network, a.k.a., constructive algorithm [5], destructive algorithm [6] and regularization algorithm. For regularization algorithm, it is to solve the problem of constrained optimization problem, remove unnecessary connections and hidden neurons in the training process, and then reduce the complexity of the model (including the structure as well as the parameters). There are two ways of constraining, i.e., constraints on the connection weights of the network, and constraints on the response of hidden neurons or constraints on the relationship between the response of hidden neurons and the target output. The method of weighting constraints includes Gaussian regularizer, Laplace regularizer, weight elimination and soft weight sharing. Among them, the most widely used is the Gaussian regularizer and the Laplace regularizer. They use the L_2-norm or L_1-norm of the connection weights as penalty functions, which are embedded in the original training process. These methods aim to achieve the sparse network topology by reducing the value of connective weights [7-9].

Above methods only consider the sparsity of the connection. In this paper, we will consider both the architecture sparsity and the representation sparsity of the neurons in deep learning. This is an effective attempt to simulate the sparse topology structure of the brain networks. We will consider the Deep Belief Network (DBN) as the deep learning model, and introduce the penalty terms on neuronal sparse connections as well as on the response of neurons into the DBN training stage to improve the energy usage and generalization capabilities of the network. By introducing the L_1-norm penalty on the connection weights and the Kullback-Leibler (KL) divergence of the hidden neurons into each layer of the stacked DBN, the sparse response of the hidden neurons and the sparse connections between different layers can be achieved together in the unsupervised training process. By such methods, the DBN can learn the fundamental weights and the effective data representation in a sparse way. Above steps thus ensure the sparse architecture of the deep network, and thus improve the compression rate, generalization capacity and discriminative accuracy of the networks.

Based on the sparse DBN network architecture, a novel feature selection method, i.e., the feature back-tracking approach based on sparse deep learning is proposed in this paper. Compared with the shallow feature selection methods, this method can understand the data in a global abstract perspective, thus find out the most crucial features of the data. Further, due to the sparsity and depth of network architecture, the important features of the data can be trace-back easily by the most discriminative abstract feature in the top layer, and an efficient implementation of the feature selection can be assured.

By applying the proposed features back-tracking method based on sparse DBN to two datasets, we can get distinct understanding on them. Mutations in individual nucleotides at the genomic level can lead to DNA sequence diversity, which may lead to human genetic disease. We used the single nucleotide polymorphism data provided by American Mind Clinical Imaging Consortium (MCIC), which had a typical high dimension (12513 SNPs loci) small sample size data (208 samples, including 92 schizophrenia patients and 116 healthy subjects), and there exists a large number of loci that are not associated with the disease. By applied the proposed features back-tracking method, 2973 pathogenic SNPs are selected from the raw SNPs data. Based on these risk loci, the diagnostic accuracy of the test data is 98.56%. In addition, several of these risk loci and their corresponding genes have been shown of great correlation with the schizophrenia in biological explanations. On the other hand, when the method is applied to the MNIST dataset. Experimental results show that with high classification accuracy, the important pixels distinguishing different handwritten digits can be finally picked out. The results show that the features back-tracking approach can identify accurately the risk loci on the genes of the mental disease as well as the key pixels of the handwritten digits, and the proposed method can also deeply improve the storage capacity as well as the search speech.

2. Unsupervised sparse learning of DBN

A DBN is a generative graphical model, composed of multiple layers of hidden units. DBN is composed of several restricted Boltzmann machines (RBMs). An RBM contains two layers, namely, the visible input layer and the hidden layer.

Let $v = (v_1, v_2, \cdots, v_{N_v})^T$ be the visible units, $h = (h_1, h_2, \cdots, h_{N_h})^T$ be the hidden units, $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_{N_v})^T$ and $\beta = (\beta_1, \beta_2, \cdots, \beta_{N_h})^T$ be the bias of the visible units and the hidden units respectively, and $W = \{W_{ij}\}_{N_v \times N_h}$ with each w_{ij} be the connection weight of v_i and h_j. (v, h) is the joint configuration of the visible and hidden units. The energy of it is described in a compact form

$$E_\theta(v, h) := E(v, h) = -\alpha^T v - \beta^T h - v^T W h$$

where $\theta = \{W, \alpha, \beta\}$. The joint probability distribution of state (v, h) is

$$P_\theta(v, h) = \frac{1}{Z_\theta} e^{-E_\theta(v,h)} \tag{1}$$

in which $Z_\theta = \sum_{v,h} e^{-E_\theta(v,h)}$ is the partition function. $P_\theta(v) = \frac{1}{Z_\theta}\sum_h e^{-E_\theta(v,h)}$ is the distribution of the observed data v_θ. The task of training an RBM is to maximizing $P_\theta(v)$, and thus to find a θ^* satisfying

$$\theta^* = \underset{\Theta}{\mathrm{argmax}}\mathcal{L}(\theta)$$

with $\mathcal{L}(\theta) = \log P_\theta(v)$. [10] proposed a faster learning using contrastive divergence (CD) to update the parameters:

$$\Delta W_{ij} = \epsilon \cdot (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon})$$

$$\Delta \alpha_i = \epsilon \cdot (\langle v_i \rangle_{data} - \langle v_i \rangle_{recon})$$

$$\Delta \beta_j = \epsilon \cdot (\langle h_j \rangle_{data} - \langle h_j \rangle_{recon})$$

where ϵ is the learning rate, and $\langle \cdot \rangle$ is the operator of expectation with the corresponding distribution denoted by the subscript.

To learn the structured network of each RBM, which helps to improve the interpretability of the network, sparse regularization can be utilized [7-9, 11-13]. Thus, some sparse regularization terms are introduced here obtain the sparsity of connections as well as the response of the neurons. The Kullback-Leibler (KL) divergence on the hidden neurons is used to achieve the response sparsity, and the L_1-norm is used on the connection weights to obtain the connective sparsity. We also add the L_2-norm on the connection weights to limit their increasing bounds. The improved objective function of the sparse RBM is given by

$$\max_{\Theta}\mathcal{L}_{new}(\Theta) = \mathcal{L}(\Theta) - \frac{1}{2}\lambda_1 ||W||_2^2 - \lambda_2 \sum_{j=1}^{N_h} KL(\rho||p_j) - \lambda_3 ||W||_1 \tag{2}$$

In which the KL divergence, $KL(\rho \parallel p_j) = \rho\log\frac{\rho}{p_j} + (1-\rho)\log\frac{1-\rho}{1-p_j}$ is the relative entropy between the two random variables with the mean ρ and the mean p_j. ρ is a sparse parameter. $p_j = \frac{1}{N_s}\sum_{q=1}^{N_s} \frac{1}{1+e^{-\sum_{i=1}^{N_v} v_i^{(q)} w_{ij} - \beta_j}}$ is the average activation probability of the j-neuron in the hidden layer with N_s samples, N_v is the node numbers of the current visual layer, λ_1 is the parameter to control the bound of W, λ_2 and λ_3 denote the penalty coefficients of the KL divergence and L_1-norm respectively.

The following is the deviation of the KL divergence.

$$\frac{\partial}{\partial W_{ij}} \sum_{j=1}^{N_h} KL(\rho \parallel p_j) = \frac{\partial}{\partial W_{ij}} \sum_{j=1}^{N_h} \left(\rho\log\frac{\rho}{p_j} + (1-\rho)\log\frac{1-\rho}{1-p_j}\right)$$

$$= \left(-\frac{\rho}{p_j} + \frac{1-\rho}{1-p_j}\right)\frac{\partial p_j}{\partial W_{ij}} = \left(-\frac{\rho}{p_j} + \frac{1-\rho}{1-p_j}\right)\frac{1}{N_s}\sum_{q=1}^{N_s} \sigma_j^{(q)}\left(1-\sigma_j^{(q)}\right)v_i^{(q)}$$

$$=\frac{1}{N_s}\left(-\frac{\rho}{p_j}+\frac{1-\rho}{1-p_j}\right)\sum_{q=1}^{N_s}\sigma_j^{(q)}\left(1-\sigma_j^{(q)}\right)v_i^{(q)}$$

here $\sigma_j^{(q)}=\sigma(\sum_{i=1}^{N_v} v_i^{(q)}W_{ij}+\beta_j)=\frac{1}{1+e^{-\sum_{i=1}^{N_v}v_i^{(q)}W_{ij}-\beta_j}}$.

Thus, the parameters of the unsupervised sparse RBM can be updated by

$$\Delta W_{ij}=\epsilon\cdot(\langle v_ih_j\rangle_{data}-\langle v_ih_j\rangle_{recon})-\lambda_1W_{ij}\ -\lambda_3\cdot sign(W_{ij})$$

$$-\lambda_2\cdot\frac{1}{N_s}\left(-\frac{\rho}{p_j}+\frac{1-\rho}{1-p_j}\right)\cdot\left(\sum_{q=1}^{N_s}\sigma_j^{(q)}\left(1-\sigma_j^{(q)}\right)v_i^{(q)}\right) \quad (3)$$

$$\Delta\alpha_i=\epsilon\cdot(\langle v_i\rangle_{data}-\langle v_i\rangle_{recon}) \quad (4)$$

$$\Delta\beta_j=\epsilon\cdot(\langle h_j\rangle_{data}-\langle h_j\rangle_{recon})-\lambda_2\cdot\frac{1}{N_s}(-\frac{\rho}{p_j}+\frac{1-\rho}{1-p_j})\cdot(\sum_{q=1}^{N_s}\sigma_j^{(q)}\left(1-\sigma_j^{(q)}\right)) \quad (5)$$

According to the above updating formula and combined with the CD sampling process, a sparse RBM and a stacked sparse DBN can be obtained.

3. Features back-tracking approach based on sparse DBN

For deep networks, the connections between two layers are on duty to encode the memories. The larger connection strength can describe the stronger relationship of the corresponding two layers. At the same time, the larger the difference of the response values of neurons between the two kinds of different data, the higher the discriminant ability of the current neurons. The two facts can be combined with the obtained sparse DBN to search for the most contributed features of the data in a back-tracking way.

The following process is the features back-tracking based on sparse DBN. Firstly, a stacked DBN with sparse architecture and sparse representation is trained by Eq. (3)-(5). Then, the learned DBN can be fine-tuned by back-propagation (BP) algorithm. Furthermore, the features back-tracking method is performed top-down to select key features by

$$I=\{i\colon|w_{ij}|\geq\alpha\ \&\ |h_{j,1}-h_{j,2}|\geq r,\exists j\in\{1,\cdots,N_h\}\} \quad (6)$$

Where w_{ij} is the connection between visible unit i and hidden unit j of the current RBM, α and r are given thresholds, $h_{j,k}$ is the mean value of the k-th class on the hidden unit j, and N_h is the unit numbers of the hidden layer. Model (6) is on two aspects. One is the larger the absolute value of w_{ij}, the higher the influence of the i-th visible unit on the j-th hidden unit. The other one is that the larger the difference of the response values between the two classes, the higher the discriminant ability of the current hidden unit. Finally, the very essential features of the data can be obtained in a back-tracking way. Each selected feature has a high probability to cause the output and make the prediction decision. The features back-tracking (FBT) approach on sparse DBN is described in Algorithm 1.

Algorithm 1. The FBT method on sparse DBN

```
Input: $L, w^{(l)}(l = L-1,\cdots,1), \alpha^{(l)}(l = L-1,\cdots,1), I^{(L)}$
Ensure: $I^{(1)}$
    For $l = L-1:-1:1$
        $I^{(l)} = i: |w_{ij}^{(l)}| \geq k^{(l)} \& |h_{j,1}^{(l+1)} - h_{j,2}^{(l+1)}| \geq r^{(l+1)}, \exists j \in I^{(l+1)}$
        $N^{(l)} = length(I^{(l)})$
    End For
Return: $I^{(1)}, N^{(l)}(l = L,\cdots,1)$
```

where L is the layer numbers of the DBN, $w^{(l)}$ is the connecting weight between the $(l+1)$-th layer and the l-th layer, $k^{(l)}$ and $r^{(l+1)}$ are thresholds and set $I^{(l)}$ contains the selected positions of the l-th layer. By back-tracking method, finally, $I^{(1)}$ is achieved, which the essential features from the data which contribute most to the final prediction.

4. Results on the SNPs data of Schizophrenia

The rapid development of genomic techniques improves the study of the relationship between mental diseases (such as schizophrenia (SZ)) and genes. Single nucleotide polymorphism (SNP) data of SZ patients is a type of genomic data resulting from genome wide association study. Each SNP is a gene sequence variation occurring commonly within a population, in which a single nucleotide - A, T, C or G - in the genome differs between paired chromosomes. The SNPs data is one typical high-dimensional data. In addition, it is extremely complex, and simple linear models have not yielded the hoped-for benefits. Many studies have investigated the crucial genes associated with SZ [14-17]. In this section, we utilize the features back-tracking method with sparse DBN to find the distinct loci of the SNPs sequence, based on which the potential risk SNPs of SZ from healthy controls can be better sought out.

The data collection was conducted by the Mind Clinical Imaging Consortium. The SNPs data contains 208 subjects including 92 schizophrenia patients and 116 healthy controls. Each SNP was represented by three numbers, 0 for $'BB'$ (no minor allele), 1 for $'AB'$ (one minor allele) and 2 for $'AA'$ (two minor alleles). The dimension for each sample is 12513.

To deal with the SNP data which are coded by 0, 1 or 2, in each RBM, they are transformed into binary sequences first. Here, the visible units $v = (v^{(1)}, v^{(2)}, v^{(3)})$, and $v^{(i)} = (v_1^{(i)}, v_2^{(i)}, \cdots, v_{N_v}^{(i)})^T$ $(i = 1,2,3)$ is an N_v-dim vector with each $v_j^{(i)} \in \{0,1\}$. For every sample, the SNP sequence is denoted as S with each $S_j \in \{0,1,2\}$. $v_j^{(i)}$ $(i = 1,2,3,$ $j = 1,2,\cdots,N_v)$ is defined as

$$v_j^{(i)} = \begin{cases} 1, & S_j = i-1 \\ 0, & S_j \neq i-1 \end{cases}$$

Then by using the unsupervised sparse learning method in Section 2, the DBN with sparse architecture and sparse representation can be achieved with training data. Additionally, based on the features back-tracking algorithm with sparse DBN, the risk SNPs loci of SZ can be selected.

The sparse DBN used here contains 5 layers, including a visible layer, three hidden layers and a decision layer. The sample size of the training data is 168, and the sample size of the testing data is 40. The number of three hidden layers we used for training the sparse DBN is 1000, 500, 200 respectively. The learning rate is 0.1 and the penalty rate is 0.0002.

The training procedure of sparse DBN and the following features back-tracking process are randomly performed 50 times. Simultaneously, based on the feature back-tracking method with sparse DBN, we use the training data to achieve the risk loci, and from the testing samples with the selected risk loci we get the average classification. The performs for the testing data before and after the feature back-tracking method are shown in Table 1. Where Dim refers to the dimension of the data, ACA is the average classification accuracy, and SCR is the space compression rate.

Table 1. Results before and after back-tracking of SNPs data

	Dim	ACA	SCR
Raw Testing Data	12513*3	0.9867	1.0000
Testing Data with Selected Risk Loci	2973	0.9856	0.0792

From the selected 2973 risk loci, there exist several SNPs that have been shown with great correlations with schizophrenia. Table 2 lists 10 typical SNPs among them.

Table 2. 10 typical selected risk loci by the feature back-tracking method

SNPs	Genes	SNPs	Genes
rs10102965	NRG1	rs1110144	CNTNAP2
rs11607732	GRIK4	rs6586002	GRID1
rs11013103	PIP4K2A	rs2765993	PIP4K2A
rs2098469	GRIN2B	rs220573	GRIN2B
rs111888901	HAAO	rs6433777	CACNB4

By http://www.genecards.org/, we can find that some genes in Table 2. have strong correlation with schizophrenia, such as NRG1, GRIK4, GRID1, PIP4K2A. In addition, GRIN2B and CNTNAP2 are related with mental retardation and neuroscience, GRIN2B is related with epileptic encephalopathy and protein-protein interactions at synapses, HAAO is present in the central nervous system, and CACNB4 is related with epilepsy.

Table 1 as well as Table 2 sustain the validation of the feature back-tracking method with sparse DBN when it is applied on the SNPs data. Firstly, the data with the selected loci nearly keep the same high classification accuracy as presented in [18]. Further, the selected risk loci can get a space saving rate about 92% with a classification accuracy around 98%, that means, nearly all of the distinguishable loci of schizophrenia from the raw data have been chosen correctly. What is the most important, several selected loci have been shown owning strong correlation with schizophrenia by biological explanation.

5. Results on MNIST data

The MNIST data contains 60,000 examples as the training set, and 10,000 examples as the test set [19]. It has 10 types handwritten digits, i.e., from 0 to 9. Each one has 784 pixels. We use the proposed FBT method to achieve key pixels of different digits.

The DBN contains 5 layers. Let $\alpha_k^{(l)} = c_k^{(l)} w_{mean}^{(l)} (l = 1, \cdots, 4)$ be a threshold of key features/sites selection with layer l for digit k, in which $c_k^{(l)}$ is a coefficient and $w_{mean}^{(l)} =$

$\frac{1}{N_v \cdot N_h}\sum_{i=1}^{N_v}\sum_{j=1}^{N_h}|w_{ij}^{(l)}|$, $\alpha_k = (c_k^{(4)} w_{mean}^{(4)}, c_k^{(3)} w_{mean}^{(3)}, c_k^{(2)} w_{mean}^{(2)}, c_k^{(1)} w_{mean}^{(1)})^T$ and $c_k = (c_k^{(4)}, c_k^{(3)}, c_k^{(2)}, c_k^{(1)})^T$. N_o and N_s are the dimension of the original data and that of the selected pixels respectively. AR_o and AR_s are the classification accuracy of the testing data and the selected key pixels of the testing data respectively. SR is the space saving rate, i.e., $SR = 1 - N_s/N_o$. AVG-AR_o and AVG-AR_s denote the average classification accuracy of the testing data and the testing data with selected sites respectively. AVG-N_s denotes the average number of the selected sites, and AVG-SR is the average space saving rate. The performance of the proposed method on all two kinds, all three kinds, and all ten kinds of digits are shown in Table 3.

Table 3. Average performance with FBT

Types of digits	AVG- AR_o	AVG- AR_s	AVG-N_s	AVG-SR
2	0.9976	0.9893	61	0.92
3	0.9975	0.9807	131	0.83
10	0.9607	0.9563	289	0.63

Table 3 shows that by the features back-tacking method combining with sparse DBN, the key pixels identifying different digits can be selected successfully. With only part of the pixels, high recognition accuracy rates can still be well kept. For some cases, the identification accuracy is even improved with the picked pixels. For example, the classification accuracy of digit 1 and digit 2 rises from 96.61% to 100% (see, Table 4 and Figure 1), the classification accuracy of digits 0, 1 and 3 increases to 100% (see Table 5 and Figure 2). The results show that the method can recognize the key positions to distinguish the digits.

Table 4. Results before and after pixel selection of digits 1 and 2

DIGIT	$\boldsymbol{N_o}$	$\boldsymbol{N_s}$	$\boldsymbol{AR_o}$	$\boldsymbol{AR_s}$	$\boldsymbol{SR}$
1	784	146	0.9974	0.9522	0.81
2	784	31	0.9961	1	0.96

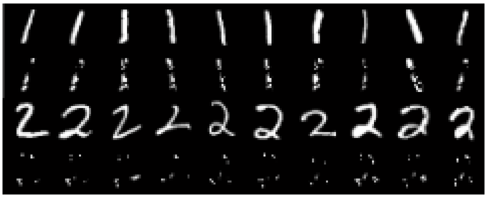

Figure 1. Original images and distinguishable images of digits 1 and 2

Table 5. Results before and after pixel selection of digits 0, 1 and 3

DIGIT	$\boldsymbol{N_o}$	$\boldsymbol{N_s}$	$\boldsymbol{AR_o}$	$\boldsymbol{AR_s}$	$\boldsymbol{SR}$
0	784	124	0.9990	0.9801	0.84
1	784	126	0.9982	1	0.84
3	784	122	1	0.9814	0.84

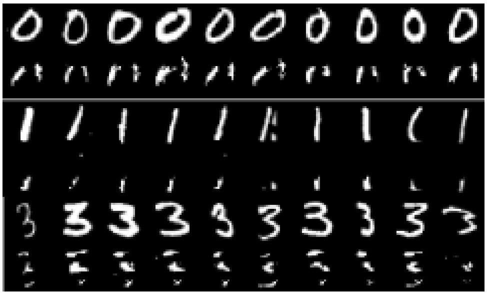

Figure 2. Original images and distinguishable images of digits 0, 1 and 3

Table 3 reveals the average space saving rate for all two kinds of digits is 92%, namely, the number of picked pixels over the data dimension is only 8%. Meanwhile, the average classification accuracy is nearly 99%. The average compression ratio of all three types digits is about 17%, with an average classification accuracy over 98%. For all ten kinds of digits, it also keeps quite high accuracy and spatial storage capacity.

We also tested the performance of one typical feature method, namely, Lasso on two kinds, three kinds, and all ten kinds of digits. Compared with the results obtained by Lasso, the proposed FBT method shows very good performance. For two kinds of digits, the AVG-N_s and AVG-AR_s obtained by Lasso are 82 and 0.9142 respectively. While, those indexes obtained by FBT are 61 and 0.9893 respectively. For three kinds of digits, the AVG-N_s and AVG-AR_s obtained by Lasso are 133 and 0.7620 respectively, and for FBT, the AVG-N_s and AVG-AR_s are 131 and 0.9807. For all ten kinds of digits, the AVG-N_s and AVG-AR_s obtained by Lasso are 213 and 0.6969 respectively, while the AVG-N_s and AVG-AR_s obtained by FBT are 289 and 0.9563. Compared with FBT, we can find that although the AVG-N_s obtained by the two methods are similar, FBT can keep quite higher classification accuracy with the selected pixels.

It usually considers MNIST data as a kind of data owning low dimensional structure. Recently, some dimensionality reduction techniques are applied to explore the lower dimensional subspace of it. While, there is no way to map high-dimensional data into low dimension and preserve all the structures of the raw data, so all the approaches try to make trade-offs, i.e., sacrificing one property to preserve another. It should be noticed that most of them are based on feature abstraction. For example, Principal Component Analysis was utilized in [20] to preserve its linear structure. In [21-24], multi-dimensional scaling and t-Distributed Stochastic Neighbor Embedding were applied to preserve its global geometry or topology structure. Deep learning approach was applied to extract its latent features [11]. Genetic algorithm was carried out to find the feature subsets [25]. Compared with the above methods, the deep learning approach is not a shallow method to find the key features, and it is no longer only a feature abstraction approach. It can directly pick out the distinct pixels of the MNIST digits from a deep layer network with a good performance of the saving ability in storage space, and it improves the classification accuracy of those in [25,26].

6. Conclusion

The interpretability of deep learning framework is quite crucial for further understanding and applying of it. Feature back-tracking method based on sparse DBN architecture provides a way for solving this issue. By simulating the sparse response of neurons for external stimulus and the sparse connection mechanism in brain system, the corresponding regularization items on the hidden neurons and the connection weights are introduced in the network learning process. Thereby, we can reduce the networks' complexity and enhance the generalization ability. By exploring the correspondence of the connections and response differences of neurons of the sparse DBN, the features back-tracking method is proposed. This method has shown quite well performance of removing irrelevant features and reducing the difficulty and complexity of learning tasks, especially in searching for risk loci of schizophrenia and picking out the intrinsic pixels of different digits.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grant No. 11471006, Grant No. 12090020 and No.12090021), the National Key Research and Development Program of China (No. 2020AAA0106302), the Science and Technology Innovation Plan of Xian (No. 2019421315KYPT004JC006) and was partly Supported by HPC Platform, Xi'an Jiaotong University.

References

[1] Colaco, S., Kumar, S., Tamang, A., Biju, V. G. "A review on feature selection algorithms." In Shetty N.Patnaik L., Nagaraj H., Hamsavath P., Nalini N. (eds) Emerging Research in Computing, Information,Communication and Applications. *Advances in Intelligent Systems and Computing.* **906** (2019) 133–153.

[2] Morris G, Nevet A, Bergman H, "Anatomical funneling, sparse connectivity and redundancy reduction in the neural networks of the basal ganglia," *Journal of Physiology-Paris.* **97** (2003) 581–589.

[3] Barlow H B, "Single units and sensation," *Representations of Vision.* 2010.

[4] Olshausen B A, Field D J, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology.* **14** (2004) 481–487.

[5] Parekh R, Yang J, Honavar V, "Constructive neural-network learning algorithms for pattern classification," *IEEE Transactions on Neural Netw.* **11** (2000) 436–451.

[6] Reed R, "Pruning algorithms-a review," *IEEE Transactions on Neural Netw.* **2** (1991) 47–55.

[7] Chen Qiao, Lan Yang, Vince D. Calhoun, Zong-Ben Xu and Yu-Ping Wang. "Sparse deep dictionary learning identifies differences of time-varying functional connectivity in brain neuro-developmental study," *Neural Networks*. **135**(2021) 91-104.

[8] Chen Qiao, Yan Shi, Yu-Xian Diao, Vince D. Calhoun and Yu-Ping Wang. "Log-sum enhanced sparse deep neural network," *Neurocomputing*. 2020, 407(24), 206-220.

[9] Chen Qiao, Bin Gao, Yan Shi. "SRS-DNN: a deep neural network with strengthening response sparsity," *Neural Computing and Applications*. 2020, 32(12), 8127-8142.

[10] Hinton G, "training products of experts by minimizing contrastive divergence," *Neural Comput.* **14** (2002) 1771–1800.

[11] Hinton G E, and Salakhutdinov R R, "Reducing the Dimensionality of Data with Neural Networks," *Science.* **313** (2006) 504–507.

[12] Hinton G E, "A Practical Guide to Training Restricted Boltzmann Machines," *Momentum.* **9** (2010) 599–619.

[13] Fischer A, and Igel C, "Training restricted Boltzmann machines: An introduction," *Pattern Recognition.* **47** (2014) 25–39.

[14] Ripke S, O'Dushlaine C, Chambert K, et al., "Genome-wide association analysis identifies 13 new risk loci for schizophrenia," *Nat Genet.* **45** (2013) 1150–1159.

[15] Luo X J, Mattheisen M, Li M, et al., "Systematic Integration of Brain eQTL and GWAS Identifies ZNF323 as a Novel Schizophrenia Risk Gene and Suggests Recent Positive Selection Based on Compensatory Advantage on Pulmonary Function," *Schizophrenia bulletin.* **41** (2015) 1294–1308.

[16] Meier L, Geer S and Bühlmann P, "The group lasso for logistic regression," *J. R. Stat. Soc. Ser. B.* **70** (2008) 53–71.

[17] Onitsuka T, Shenton M E, Salisbury D F, et al., "Middle and inferior temporal gyrus gray matter volume abnormalities in chronic schizophrenia: an MRI study," *Am. J. Psychiatry.* **161** (2004) 1603-1611.

[18] Qiao C, Lin D D, Cao S L, et al., "The effective diagnosis of schizophrenia by using multi-layer RBMs deep networks," *IEEE International Conference on Bioinformatics and Biomedicine.* 2015, 603–606.

[19] LeCun, Y., Cortes, C, "MNIST handwritten digit database," 2010. URL http://yann.lecun.com/exdb/mnist.

[20] Jiang B, Ding C, Luo B, et al, "Graph-Laplacian PCA: Closed-Form Solution and Robustness," *CVPR.* **9** (2013) 3492–3498.

[21] Silva V D, Tenenbaum J B, "Sparse Multidimensional Scaling using Landmark Points," *Technical report.* Stanford University, 2004.

[22] Dzwinel W, Wcisło R, "Very Fast Interactive Visualization of Large Sets of High-dimensional Data," *Procedia Computer Science.* **51** (2015) 572–581.

[23] Maaten L, Hinton G, "Visualizing data using t-SNE," *Journal of Machine Learning Research.* **9** (2008) 2579–2605.

[24] Lee J A, Verleysen M, "Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants," *Procedia Computer Science.* **4** (2011) 538–547.

[25] Stefano C D, Fontanella F, Marrocco C, et al., "A GA-based feature selection approach with an application to handwritten character recognition," *Pattern Recognition Letters.* **35** (2014) 130–141.

[26] Wang J, Wonka P, Ye J, "Lasso screening rules via dual polytope projection," *Journal of Machine Learning Research.* **16** (2015) 1063–1101.