

Research Article

Bidirectional Nonnegative Deep Model and Its Optimization in Learning

Xianhua Zeng, Zhengyi He, Hong Yu, and Shengwei Qu

Chongqing Key Lab of Computational Intelligence, College of Computer Science and Technology,
Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Xianhua Zeng; xianhuazeng2005@163.com

Received 18 August 2016; Accepted 17 October 2016

Academic Editor: Tongliang Liu

Copyright © 2016 Xianhua Zeng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nonnegative matrix factorization (NMF) has been successfully applied in signal processing as a simple two-layer nonnegative neural network. Projective NMF (PNMF) with fewer parameters was proposed, which projects a high-dimensional nonnegative data onto a lower-dimensional nonnegative subspace. Although PNMF overcomes the problem of out-of-sample of NMF, it does not consider the nonlinear characteristic of data and is only a kind of narrow signal decomposition method. In this paper, we combine the PNMF with deep learning and nonlinear fitting to propose a bidirectional nonnegative deep learning (BNDL) model and its optimization learning algorithm, which can obtain nonlinear multilayer deep nonnegative feature representation. Experiments show that the proposed model can not only solve the problem of out-of-sample of NMF but also learn hierarchical nonnegative feature representations with better clustering performance than classical NMF, PNMF, and Deep Semi-NMF algorithms.

1. Introduction

In the study of machine learning, pattern recognition, computer vision, and image processing, it is an important problem to find the effective representations of the input data matrix with nonnegative elements and very high dimensions. In 1999, Lee and Seung had proposed a classical feature representation method, named nonnegative matrix factorization (NMF) [1], which effectively solved the above problems. The basic idea and analysis of the NMF algorithm may be simply described as follows.

Given a nonnegative data matrix $X \in R_+^{m \times n}$, which is a collection of n samples as columns, and each sample is nonnegative so that they allow only additive not subtractive and linear combinations. To a degree, it can capture the essence of intelligent data description. And the objective function can be defined as

$$\min_{W \geq 0, H \geq 0} \|X - WH^T\|. \quad (1)$$

Although the NMF is optimal for learning the parts of objects, it suffers from the out-of-sample problem [2, 3]; namely, it is indirect or repeats the factorization to obtain

the coefficients of any new coming examples. To overcome the disadvantages of the NMF, after that, the researchers put forward some improved methods based on the NMF algorithm. For example, Yuan et al. proposed a Projective NMF (PNMF) [4] in 2009. The PNMF is a modified form of the traditional NMF, with strong sparseness and orthogonality [4, 5] under the projection assumption. It only needs to calculate a nonnegative matrix W , thereby reducing the amount of computation at each iteration; that is, the PNMF learns a nonnegative matrix to directly project X onto the lower-dimensional nonnegative subspace. If W denotes the basis matrix, the PNMF treats $Y = W^T X$ as the coefficient and utilizes $WW^T X$ to reconstruct X . So its objective function is

$$\min_{W \geq 0} \|X - WW^T X\|. \quad (2)$$

The PNMF has fewer parameters than the NMF, and it is widely used in linear dimension reduction and can solve the problem about out-of-sample deficiency. Being the same with the NMF, the PNMF is a linear dimensionality reduction method, but many data present the nonlinear characteristics [6]. At the same time, the NMF and the PNMF only factorize the original data one time [7]. In many situations, the

nonnegative data sampled from real applications are usually very complex and need to be factorized many times for obtaining the high-level deep features with distinction and strong representation ability. Some studies have shown that, in order to learn the high-level representations of complex data and have better performance in image understanding and speech perception, the deep learning is needed [6]. And the deep learning has a profound impact both in academia and in industry fields since Hinton and Salakhutdinov published a known article [8] in Science in 2006. This article shows the following: (1) artificial neural network with a lot of hidden layers has excellent ability for learning characteristics, which is more essential to describe data and facilitates the visualization, clustering, and classification; (2) the difficulty on training the deep neural network can be overcome by the “layer by layer initialization” (layer-wise pretraining). With the success of training deep architectures, several variants of deep learning have been introduced [6, 9]. These multilayer algorithms take hierarchical approaches in feature extraction and provide efficient solution to complex problems, and they use an error backpropagation algorithm and unsupervised learning to obtain an effective representation model. However, they have not considered the following concerns: (1) the weights should be nonnegative when a lot of physical signal is nonnegative data; (2) the pure additive description uses little component to make the components of the nonnegative data clear.

For obtaining the deep nonnegative feature representation, Trigeorgis et al. applied the concept of Semi-NMF [10] to propose a Deep Semi-NMF [9] that is able to learn hidden deep representations of the original data. In the Semi-NMF, the goal is to construct a low-dimensional nonnegative representation H^+ of our original data X^\pm , with the bases matrix Z^\pm serving as the mapping between our original data and its lower-dimensional representation [10]. The Deep Semi-NMF model finds a representation of the data that has a similar interpretation at the top layer. The input data matrix is now further analyzed as a product of multiple factors $Z^\pm = Z_1^\pm Z_2^\pm Z_3^\pm \cdots$, which are thought to be deep seminonnegative matrix factorization. That means it is able to decompose the data in n different ways according to multiple different attributes:

$$\begin{aligned} H_i^+ &= Z_{i+1}^\pm H_{i+1}^+, \quad i = 1, 2, \dots, n-1, \\ X^\pm &= \prod_{i=1} Z_i^\pm H_k^+, \quad k = 1, 2, \dots, n. \end{aligned} \quad (3)$$

Although the Deep Semi-NMF uses a multilayer model to obtain more features, it can only deal with seminonnegative data, which is a linear transformation with weak representation capacity. Moreover, the Deep Semi-NMF model still has the out-of-sample problem.

Based on the above analysis, the PNMF only computes one projection matrix and it cannot learn more rich features, especially when the data are a nonlinear or near a nonlinear manifold, or the data are hierarchically generated. Motivated by the ideas of the PNMF, the Deep Semi-NMF, and deep learning (especially, AutoEncoder [8, 11]), in this paper, we

propose a novel model which we call bidirectional nonnegative deep learning (BNDL), for learning more helpful and meaningful deep nonnegative representations of the original data with nonlinear characteristic and overcoming the out-of-sample problem. In Section 2, we introduce our BNDL method and the analysis of the optimal objective functions. And we give the corresponding algorithms in Section 3. Experiments are demonstrated in Section 4. In Section 5, we briefly give some conclusion remarks about this paper.

2. Bidirectional Nonnegative Deep Learning Model

2.1. Motivation. The particular attraction of the NMF algorithm is the nonnegative constraints, and it is useful for data representation in clustering. But the NMF is a simple linear coding algorithm using a single layer network with nonnegative constraints, and it suffers from the out-of-sample deficiency which cannot directly obtain the codes of any new coming examples [12, 13].

To the PNMF algorithm, it uses the transpose matrix of the learned basis matrix as the projection matrix, which obtains nonnegative coefficients for any new coming examples [4, 14]. Although it overcomes the problem of out-of-sample of the NMF, the PNMF is also a linear coding algorithm and simple single layer decomposition.

On the other hand, the current existing deep network models rarely consider the nonnegative constraints, even if the newest related model Deep Semi-NMF [9] only broadens an incomplete nonnegative constraint and is still a linear model.

In this paper, we propose a nonnegative hierarchical data representation model, named bidirectional nonnegative deep learning (BNDL) model, which applies the concept of PNMF to train an initial multilayer nonlinear structure that is able to learn hidden complete deep representations of the original data.

Different from the other deep architectures, the BNDL firstly constructs a pretraining deep network through stacking every nonnegative two layers network independently to get the whole network, and the learning process of each layer is to combine the PNMF and a designed nonlinear mapping. That is to say that each time we do one-step decomposition, then the basis matrix of two-layer BNDL can be regarded as the weight matrix of the deep network, and the output of this step can be used as the input of the next layer by a Sigmoid function. Upwards, iterating this process, we can get a deep network. Downwards, we can reconstruct the original sample data. Because BNDL only learn one layer in each step, we can fast build a deep network. The hierarchical feature extraction strategy learns more meaningful, helpful features and higher-order nonnegative nonlinear characteristics than one-step learning. Finally, a fine-tune training is applied to improve the reconstruction performance and deep features of our deep network under the nonnegative weight value constraints.

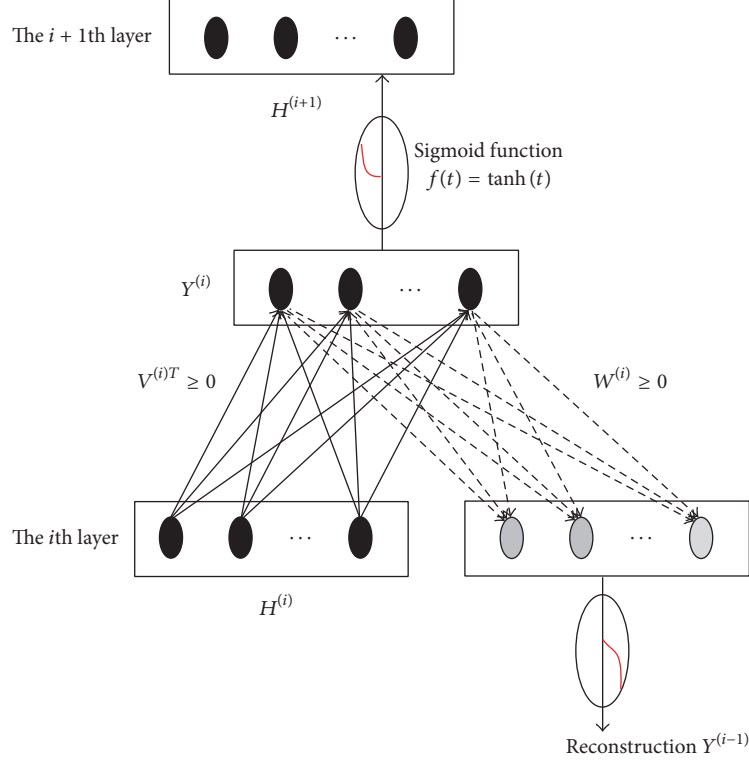


FIGURE 1: The two-layer binetwork structure for BNDL.

2.2. Bidirectional Nonnegative Deep Learning Model. Let $X = [X_1, X_2, \dots, X_n]$ denote the data sample set, among which $X_i \in R^m$ denotes the feature descriptor of the i th sample and n is the number of total samples. Here we assume that the input data matrix X is nonnegative. Let k denote the dimension of the desired dimension-reduced feature space. The task of data factorization is to get a nonnegative basis matrix $W = [w_1, w_2, \dots, w_k] \in R^{m \times k}$ and its corresponding coefficient matrix $Y = V^T X$. Here $V \in R_{k \times m}$ devotes the projection matrix that transforms an m -dimensional feature vector into a k -dimensional feature space. The matrices $H^{(i)}$, $i = 1, \dots, I$, are the output matrices of the first $i - 1$ layers, and $H^{(1)}$ is equal to the original matrix X . The objection of the i th factorization is as close as possible $H^{(i)}$; that is, $H^{(i)} \approx W^{(i)} Y^{(i)}$. So the objection function for projective nonnegative multilayer factorization can be defined as

$$\begin{aligned} \min_{V^{(i)}, W^{(i)}} & \quad \|H^{(i)} - W^{(i)} Y^{(i)}\|^2 \\ \text{S.t.} & \quad Y^{(i)} = V^{(i)T} H^{(i)} \\ & \quad H^{(i+1)} = \tanh(C * Y^{(i)}) \\ & \quad W^{(i)} \geq 0, H^{(i)} \geq 0, V^{(i)} \geq 0, \end{aligned} \quad (4)$$

where C is a positive factor to avoid the too large input amount for Sigmoid function and $W^{(i)} \geq 0$ (or $H^{(i)} \geq 0$ or $V^{(i)} \geq 0$) denotes that each element of it is nonnegative.

Due to preserving the same S-shape nonlinear mapping function of the top-to-bottom operator, the top-to-bottom reconstruction basis $W^{(i)}$ should also be constrained to reconstruct the input $Y^{(i-1)}$ of the $(i - 1)$ th layer. So the new objective cost is further improved into

$$\begin{aligned} \min_{V^{(i)}, W^{(i)}} & \quad \left\{ \|H^{(i)} - W^{(i)} Y^{(i)}\|^2 + \lambda \|Y^{(i-1)} - f(W^{(i)} Y^{(i)})\|^2 \right\} \\ \text{S.t.} & \quad Y^{(i)} = V^{(i)T} H^{(i)} \\ & \quad H^{(i+1)} = f(C * Y^{(i)}) \\ & \quad W^{(i)} \geq 0, H^{(i)} \geq 0, V^{(i)} \geq 0 \\ & \quad f(x) = \tanh(x), \end{aligned} \quad (5)$$

where λ is the balance factor. In our experiments, if $i = 1$, $\lambda = 0$; else $\lambda = 1$.

So the two-layer structure for constructing BNDL can be illustrated in Figure 1.

2.3. Multiplicative Update Rule for Initializing the Bottom-to-Up Weight of BNDL. For the simplicity of initializing the bottom-to-up weights $V^{(i)}$, we do not firstly consider the second term of the objective function in (5), so the generative weights $V^{(i)}$ and reconstructive weights $W^{(i)}$ are only computed by optimizing the first term. The simple objective function is

$$\min_{W \geq 0, V \geq 0} \left\{ \|H^{(i)} - W^{(i)} V^{(i)T} H^{(i)}\|^2 \right\}. \quad (6)$$

Obviously,

$$\begin{aligned}\Gamma_{\text{BNDL}} &= \left\| H^{(i)} - W^{(i)} V^{(i)T} H^{(i)} \right\|^2 \\ &= \text{tr} \left(\left(H^{(i)} - W^{(i)} V^{(i)T} H^{(i)} \right) \left(H^{(i)} - W^{(i)} V^{(i)T} H^{(i)} \right)^T \right) \\ &= \text{tr} \left(H^{(i)} H^{(i)T} \right) - 2 \text{tr} \left(W^{(i)} V^{(i)T} H^{(i)} H^{(i)T} \right) \\ &\quad + \text{tr} \left(W^{(i)} V^{(i)T} H^{(i)} H^{(i)T} V^{(i)} W^{(i)T} \right).\end{aligned}\quad (7)$$

According to minimizing (7), simultaneously solving $V^{(i)}$ and $W^{(i)}$ is a NP-difficult problem. So the scalable solution of (6) is alternatively optimizing with respect to $V^{(i)}$ and $W^{(i)}$ by fixing one of them.

When fixing the $V^{(i)}$, the objective function solved by Lagrangian multiplier method [9] is transformed into

$$\begin{aligned}\Gamma_{\text{BNDL}} &= \text{tr} \left(H^{(i)} H^{(i)T} \right) - 2 \text{tr} \left(W^{(i)} V^{(i)T} H^{(i)} H^{(i)T} \right) \\ &\quad + \text{tr} \left(W^{(i)} V^{(i)T} H^{(i)} H^{(i)T} V^{(i)} W^{(i)T} \right) \\ &\quad - \text{tr} (\phi W),\end{aligned}\quad (8)$$

where ϕ is the Lagrangian multiplier, in which the constraints W and H are nonnegative.

The minimization of (8) is equal to letting the equation

$$\begin{aligned}\frac{\partial \Gamma_{\text{BNDL}}}{\partial W^{(i)}} &= -2V^{(i)T} H^{(i)} H^{(i)T} + 2W^{(i)} V^{(i)T} H^{(i)} H^{(i)T} V^{(i)} \\ &\quad - \phi = 0.\end{aligned}\quad (9)$$

Under the nonnegative condition constraints

$$\begin{aligned}W^{(i)} &\geq 0, \\ \phi &\geq 0.\end{aligned}\quad (10)$$

Based on the KKT conditions [15], the minimum solution of (8) is satisfied as

$$W_{jk} \phi_{jk} = 0, \quad (11)$$

where W_{jk} is the item at the j th row and the k th column of W . We can get the following equality via combining (10) and (11) with respect to the parameters W .

$$\begin{aligned}\left\{ -2V^{(i)T} H^{(i)} H^{(i)T} + 2W^{(i)} V^{(i)T} H^{(i)} H^{(i)T} V^{(i)} \right\}_{jk} W_{jk} \\ = 0.\end{aligned}\quad (12)$$

To (12), moving the negative items to the right-hand side in (12), we have

$$\begin{aligned}\left\{ W^{(i)} V^{(i)T} H^{(i)} H^{(i)T} V^{(i)} \right\}_{jk} W_{jk} \\ = \left\{ V^{(i)T} H^{(i)} H^{(i)T} \right\}_{jk} W_{jk}.\end{aligned}\quad (13)$$

So we can get a Multiplicative Update Rule (MUR) from (4) for any two-layer learning of BNDL:

$$W^{(i)} \leftarrow \frac{V^{(i)T} H^{(i)} H^{(i)T}}{W^{(i)} V^{(i)T} H^{(i)} H^{(i)T} V^{(i)}} \odot W^{(i)}, \quad (14)$$

where \odot denotes the dot product, that is, the product of the corresponding elements from two matrices, and the matrix division is the dot division of the corresponding elements.

Similarly, the MUR [12] for V is

$$V^{(i)} \leftarrow \frac{W^{(i)T} H^{(i)} H^{(i)T}}{W^{(i)T} W^{(i)} V^{(i)} H^{(i)} H^{(i)T}} \odot V^{(i)}. \quad (15)$$

Moreover, by referring the proof of Theorem 1 in the literature [12], we can prove the following conclusion, that is, Theorem 1. Note that the proof of Theorem 1 is similar to the literature [12]; due to the limited space, we omit the detailed proof and derivations.

Theorem 1. *The Euclidean metric $\|H^{(i)} - W^{(i)} V^{(i)T} H^{(i)}\|^2$ is nonincreasing under the following two Multiplicative Update Rules:*

$$\begin{aligned}W^{(i)} &\leftarrow \frac{V^{(i)T} H^{(i)} H^{(i)T}}{W^{(i)} V^{(i)T} H^{(i)} H^{(i)T} V^{(i)}} \odot W^{(i)}, \\ V^{(i)} &\leftarrow \frac{W^{(i)T} H^{(i)} H^{(i)T}}{W^{(i)T} W^{(i)} V^{(i)} H^{(i)} H^{(i)T}} \odot V^{(i)}.\end{aligned}\quad (16)$$

And there must exist the optimal $W^{(i)}$ and $V^{(i)}$ to make the value of (7) achieve a stationary stable point.

2.4. The Modification of the Top-to-Bottom Reconstruction Weight $W^{(i)}$. After decomposing the input $H^{(i)}$ to get non-negative generative weights $V^{(i)}$, fixing the first term and minimizing the second term of the objective function (5), we need to modify reconstruction weights $W^{(i)}$ for obtaining the optimal top-to-bottom nonlinear mapping to the $(i-1)$ th layer input; that is, $Y^{(i-1)} \approx f(W^{(i)} Y^{(i)})$, $i \geq 2$. So we can easily get the following theorem.

Theorem 2. *For minimizing the square objective function $\|Y^{(i-1)} - f(W^{(i)} Y^{(i)})\|^2$, because the Sigmoid functions $f(\cdot)$ for neuron outputs are a monotonic function, the optimal solution of $W^{(i)}$ is*

$$W^{(i)} = \left(f^{-1} \left(Y^{(i-1)} \right) Y^{(i)T} \left(Y^{(i)} Y^{(i)T} \right)^+ \right)_+, \quad (17)$$

where $(\cdot)^+$ is the pseudo-inverse computation and $(\cdot)_+$ is to preserve nonnegative elements and substitute the negative elements into zero.

Corollary 3. *If the Sigmoid transfer function for nonnegative neural networks is the hyperbolic tangent function $f(x) = (e^x - e^{-x}) / (e^x + e^{-x})$, then the inverse function about the function value $Y^{(i-1)} \in (0, 1)$*

$$f^{-1} \left(Y^{(i-1)} \right) = \frac{1}{2} \ln \frac{1 + Y^{(i-1)}}{1 - Y^{(i-1)}}. \quad (18)$$

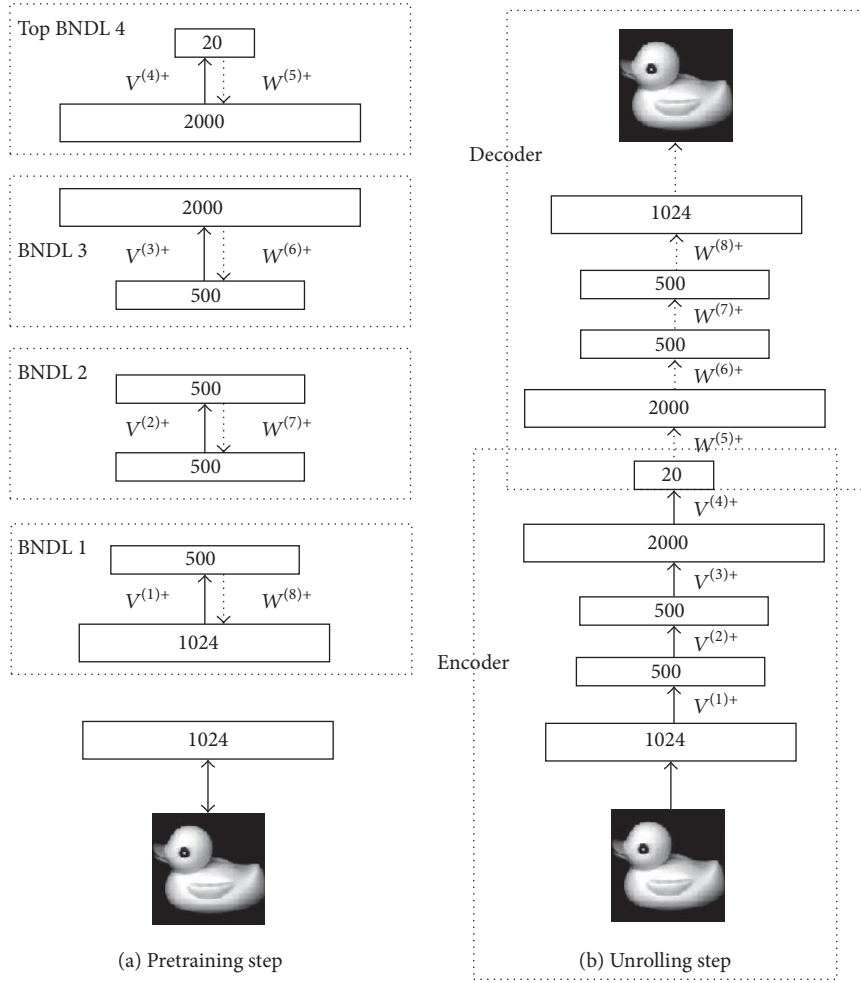


FIGURE 2: The unrolling illustration of BNDL.

For obtaining the stability in our deep model, the input signals are normalized into the interval $[0, 1]$. Obviously the reconstruction signals should lie in the interval $[0, 1]$, so we also introduce the other common nonlinearity for neural networks, the Rectified Linear Unit (ReLU) [15].

Corollary 4. *If the transfer function for nonnegative neural networks is the ReLU function at $[0, 1]$, that is, $f(x) = \min\{\max(0, x), 1\}$, then the inverse function about the function value $Y^{(i-1)} \in (0, 1)$ has the same representation; that is,*

$$f^{-1}(Y^{(i-1)}) = \min\{\max\{0, Y^{(i-1)}\}, 1\}. \quad (19)$$

2.5. Unfolding Each Two-Layer Network to Construct a Multilayer Nonnegative Network. Though the weight matrices of all layers are learned, they are only efficient in each two-layer network, and they are not optimal for the whole network. The weights in higher layers may be not optimal for the lower layers [16, 17]. So after greedily learning good initial values for the weights in every two layers, we unroll each two-layer nonnegative network by using the $W^{(i)}$ and $V^{(i)}$ to construct a multilayer nonnegative nonlinear network which

can be seen in Figure 2. From top to down, we hope that the reconstruction error is as small as possible. So the BP algorithm is applied to fine-tune the unrolled deep networks under the nonnegative-weight constraints.

3. The BNDL Algorithm Description

Summarizing the above analysis, unrolling each two-layer nonnegative network, the bottom-to-top weights $V^{(i)}$ connect each nonlinear neuron layer to get the first multilayer structure; then the top-to-bottom weights $W^{(i)}$ connect each nonlinear neuron layer from the top layer to the bottom layer (i.e., the input layer) and get a whole multilayer nonlinear nonnegative deep network structure as illustrated in Figure 2. For further optimizing the multilayer deep networks, fine-tuning all weights by the improved BP algorithm reduces the reconstruction error and ensures the weights to be nonnegative by constraining the gradient descent iteration $W(t) + \Delta W \geq 0$. In our fine-tune stage, the weights are updated by using the conjugate gradient with three-time linear searching from the source codes in literature [8], and

TABLE 1: The characteristics of the datasets.

Datasets	Dimension (m)	Total number (n)	Number of classes
COIL20	1024	1440	20
COIL100	1024	7200	100
CMU PIE	1024	2856	68

nonnegative weights are ensured by the computation $W(t+1) = \max\{0, W(t) + \Delta W\}$.

The main steps of the BNDL algorithm are described in Algorithm 1.

Algorithm 1 (training algorithm for BNDL). Learning algorithm for BNDL includes the following steps.

Step 1. Pretrain bottom-top initial nonnegative weights $W^{(i)}$ and $V^{(i)}$ for each layer.

- (1) Initialize the first layer $H^{(0)} = X$, getting $W^{(0)} \in R^{m \times k}$ and $V^{(0)} \in R^{m \times k}$ with nonnegative random values at interval $[0, 1]$.
- (2) Repeatedly update the reconstructed weights $W^{(i)}$ and generative weights $V^{(i)}$ by, respectively, using the iterative formulas (14) and (15) until the maximum number of iterations.
- (3) Modify the top-to-bottom reconstruction weight $W^{(i)}$ by (17).

Step 2. Unroll each two-layer nonnegative network by using the $W^{(i)}$ and $V^{(i)}$ to construct a multilayer nonnegative nonlinear network as shown in Figure 2.

Step 3. Fine-tune weights $V^{(i)}$ and $W^{(i)}$ by the improved BP algorithm under the constraints $V(t) + \Delta V \geq 0$ and $W(t) + \Delta W \geq 0$. The updated nonnegative weights are performed by using the conjugate gradient of the reconstruction error with three-time linear searching from the source codes in literature [8], and $W(t+1) = \max\{0, W(t) + \Delta W\}$.

Step 4. Run the K -means algorithm, the feature representation by the given nonnegative output.

4. Experiments

In this section, we carry out some experiments to verify the validity of BNDL on three datasets including COIL20, COIL100, and CMU PIE, as shown in Table 1. In order to compare the clustering experimental results, we use the Accuracy (AC) and Normalized Mutual Information (NMI) [13] as the evaluation measures.

4.1. Datasets Introduction. COIL20 dataset [17] in this experiment contains 20 objects. The images of each object are taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. The size of each image is preprocessed into 32×32 pixels, with 256 grey levels per pixel. Thus, each image is represented by a 1024-dimensional vector.

COIL100 dataset [17] which contains 100 objects is used to the experiment. The images of each object are taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. The size of each image is preprocessed into 32×32 pixels, with 256 grey levels per pixel. Thus, each image is represented by a 1024-dimensional vector.

CMU PIE [18] includes the original database of 41,368 images of 68 people, each person under different poses and different illumination conditions and with different expressions. Due to the limitation of experiment platform, in this experiment, the fore 42 face images corresponding to each person are extracted from the preprocessed CMU PIE dataset [18] with 11,544 face pose images of 68 persons at size 32×32 pixels, so the experimental data includes 2856 face images of 68 persons where each person corresponds to 42 face images with different poses.

4.2. Clustering Experiment on COIL20, COIL100, and CMU PIE Datasets. The related literatures have demonstrated that NMF method has good performance for clustering, especially in the image clustering task. So we mainly do experiments about clustering by comparing the most related methods including NMF [1], PNMF [2], Deep Semi-NMF [9], and our deep learning model, that is, BNDL. In this experiment, the number of layers about BNDL is set to five, the number of nodes in its first layer is the dimension of the input data, the second layer has 500 nodes, the third layer has 500 nodes, the fourth layer has 2000 nodes, and the last has the same number with classes k . So the whole network node is 1024-500-500-2000- k . Note that fine-tuning training is carried out on nine-layer networks. The network structure can be seen in Figure 2.

In this subsection, “2nd layer” expresses the second layer of the deep network matrix decomposition; “rec. 1st layer” expresses the first reconstruction layer of the deep network matrix reconstruction, and so on.

In the clustering experiment, we use the feature from the all learned layers and reconstruction layers of BNDL to compare with other methods. All algorithms use the same iterations (5000 times), the same initialization method (random initialization), and the same termination conditions (the error is less than 10^{-6}). The clustering performance based on AC and NMI is shown in Table 2.

From the results of Table 2, we can see that BNDL has the following advantages:

- (1) Compared with the single layer matrix decomposition network, BNDL learns more rich features while the clustering effect is not reduced.
- (2) Compared with the deep network matrix decomposition network, each layer of BNDL has better clustering effect.
- (3) Each layer of BNDL has a stable clustering performance, which is better for the data representation and downward transmission.

TABLE 2: Performance comparison on different datasets.

Algorithms	COIL20		COIL100		CMU PIE	
	AC	NMI	AC	NMI	AC	NMI
NMF	0.6069	0.7216	0.4483	0.7236	0.4321	0.7197
PNMF	0.6340	0.7356	0.4701	0.7409	0.1796	0.4326
Deep Semi-NMF 2nd layer	0.2889	0.4124	0.3114	0.5502	0.6218	0.7868
Deep Semi-NMF 3rd layer	0.3688	0.4304	0.4064	0.6695	0.6047	0.7720
Deep Semi-NMF 4th layer	0.4375	0.5131	0.4065	0.6367	0.5018	0.7313
Deep Semi-NMF 5th layer	0.6063	0.7141	0.4018	0.6674	0.8074	0.9402
<i>BNDL 2nd layer</i>	<i>0.5687</i>	<i>0.7275</i>	<i>0.4624</i>	<i>0.7389</i>	<i>0.2843</i>	<i>0.5880</i>
<i>BNDL 3rd layer</i>	<i>0.5806</i>	<i>0.7277</i>	<i>0.4754</i>	<i>0.7273</i>	<i>0.3459</i>	<i>0.6495</i>
<i>BNDL 4th layer</i>	<i>0.6431</i>	<i>0.7434</i>	<i>0.3890</i>	<i>0.6441</i>	<i>0.3186</i>	<i>0.5754</i>
<i>BNDL 5th layer</i>	<i>0.6458</i>	<i>0.6760</i>	<i>0.4499</i>	<i>0.6895</i>	<i>0.3533</i>	<i>0.6393</i>

TABLE 3: Performance comparison of reconstruction data on different datasets.

Algorithms	COIL20		COIL100		CMU PIE	
	AC	NMI	AC	NMI	AC	NMI
NMF rec.	0.6267	0.7490	0.4468	0.7200	0.4391	0.7569
PNMF rec.	0.6528	0.7630	0.4686	0.7373	0.1866	0.4698
Deep Semi-NMF rec. 1st layer	0.5563	0.6304	0.3689	0.5783	0.8277	0.9506
Deep Semi-NMF rec. 2nd layer	0.5535	0.6590	0.4125	0.6471	0.7962	0.9156
Deep Semi-NMF rec. 3rd layer	0.6708	0.7263	0.4169	0.6568	0.8319	0.9447
Deep Semi-NMF rec. 4th layer	0.5271	0.5701	0.3114	0.5262	0.5963	0.7627
<i>BNDL rec. 1st layer</i>	<i>0.5903</i>	<i>0.6793</i>	<i>0.5004</i>	<i>0.7313</i>	<i>0.3771</i>	<i>0.6503</i>
<i>BNDL rec. 2nd layer</i>	<i>0.5813</i>	<i>0.6819</i>	<i>0.4760</i>	<i>0.7424</i>	<i>0.3074</i>	<i>0.6074</i>
<i>BNDL rec. 3rd layer</i>	<i>0.6132</i>	<i>0.6815</i>	<i>0.4753</i>	<i>0.7328</i>	<i>0.2532</i>	<i>0.5640</i>
<i>BNDL rec. 4th layer</i>	<i>0.6049</i>	<i>0.6827</i>	<i>0.4826</i>	<i>0.7426</i>	<i>0.2454</i>	<i>0.5377</i>

- (4) From the results on COIL20 and COIL100, we can see that BNDL has a better clustering performance for large scale samples, which is more conducive to the characteristic expression of complex data.
- (5) Experimental results on CMU PIE face images are disappointed in a degree. Compared with our model, NMF and Deep Semi-NMF suffer from the out-of-sample problem. But both our BNDL model and the classical PNMf can solve the problem about out-of-sample deficiency. Moreover, our BNDL obtains the better performance than the PNMf on CMU PIE face dataset. In addition, face clustering and classification usually get the better results by using cosine distance metric, so the graph regularization with the sine similarity is introduced to improve the BNDL in the future.

The reconstruction performance of BNDL is also excellent. In order to compare the reconstruction performance of each algorithm, here we compare the clustering performance of reconstruction data. Table 3 is the results on the three datasets. It can be found that the final reconstruction result of BNDL does not decrease, compared with the reconstruction result of the single layer matrix decomposition. And BNDL is better than other deep network matrix decomposition. The experiment results show that the reconstruction data

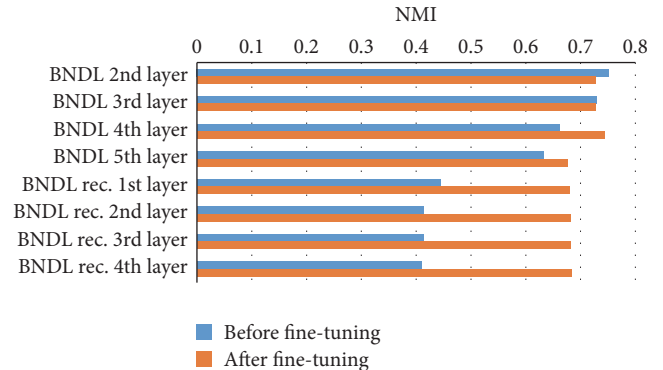


FIGURE 3: NMI of before and after fine-tuning of BNDL on COIL20.

of BNDL has better clustering effect, and the deep network structure is more conducive to the multilayer feature expression.

The reason why BNDL has better reconstruction results is that it connects each layer into a deep learning network after decomposing all layers and fine-tuning the whole network by improved BP algorithm to reduce reconstruction error to minimum. To verify the effect of fine-tuning to the whole network, we compare the difference of before fine-tuning and after fine-tuning, as shown in Figures 3 and 4. Two figures

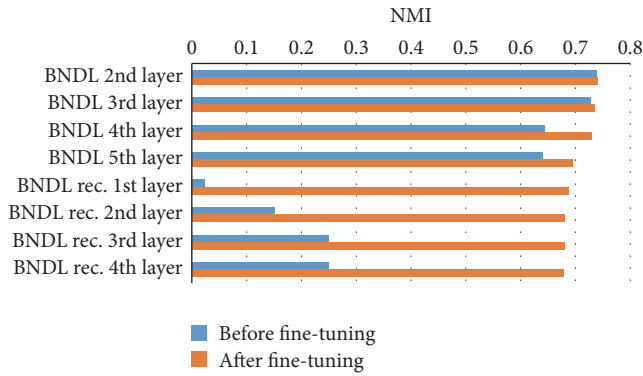


FIGURE 4: NMI of before and after fine-tuning of BNDL on COIL100.

imply that the clustering ability of BNDL has a considerable improvement after fine-tuning, which means fine-tuning has a strong effect on network; that is to say, the multilayer feature representation ability of network is strongly boosted after fine-tuning.

5. Conclusion

This paper proposes a bidirectional nonnegative deep learning model to obtain effective feature representation, which can automatically learn a deep hierarchy with nonlinear and nonnegative feature representations via inputting a given nonnegative dataset, and such representations are demonstrated to be suited for clustering. About the PNMF, it is a linear dimensionality reduction and feature representation method, and it only factorizes the original data in one step. And the Deep Semi-NMF can construct a multitime linear factorization to learn more features, but it is still a linear dimensionality reduction method and not an absolutely nonnegative treatment. Our BNDL model combines the advantages of PNMF and deep belief networks under the inspiration of the Deep Semi-NMF and overcomes the above-mentioned shortcomings. At the same time, we designed an effective learning algorithm for optimizing the corresponding parameters of our BNDL model. Lastly, we show its better clustering performance compared with the single-layered NMF, PNMF, and Deep Semi-NMF to a degree. In addition, our method avoids the out-of-sample problem and negative feature representation in the different motivation with Deep Semi-NMF.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61672120, 61379114) and the Chongqing Natural Science Foundation Program (Grant no. cstc2015cyjA40036).

References

- [1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [2] Z. Yuan and E. Oja, "Projective non-negative matrix factorization for image compression and feature extraction," in *Proceedings of the Image Analysis*, pp. 333–342, Springer, Berlin, Germany, June 2005.
- [3] Y. Bengio, J. F. Paiement, P. Vincent, D. Olivier, L. Nicolas, and O. Marie, "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering," in *Advances in Neural Information Processing Systems*, vol. 16, pp. 177–184, 2004.
- [4] Z. Yuan, Z. Yang, and E. Oja, *Projective Non-Negative Matrix Factorization: Sparseness, Orthogonality, and Clustering*, Neural Processing Letters, 2009.
- [5] N. Guan, X. Zhang, Z. Luo, D. Tao, and X. Yang, "Discriminant projective non-negative matrix factorization," *PLoS ONE*, vol. 8, no. 12, Article ID e83291, 12 pages, 2013.
- [6] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–27, 2009.
- [7] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.
- [8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [9] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller, "A Deep Semi-NMF model for learning hidden representations," in *Proceedings of the 31st International Conference on Machine Learning (ICML '14)*, pp. 3677–3688, Beijing, China, June 2014.
- [10] C. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2010.
- [11] Y. Bengio, P. Lamblin, D. Popovici, and L. Hugo, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems*, vol. 19, pp. 153–160, 2007.
- [12] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 556–562, 2001.
- [13] S. Z. Li, X. W. Hou, H. J. Zhang, and Q. S. Cheng, "Learning spatially localized, parts-based representation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 1-207–1-212, Kauai, Hawaii, USA, December 2001.
- [14] X. Liu, S. Yan, and H. Jin, "Projective nonnegative graph embedding," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1126–1137, 2010.
- [15] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [16] A. Cichocki and R. Zdunek, "Multilayer nonnegative matrix factorization using projected gradient approaches," *International Journal of Neural Systems*, vol. 17, no. 6, pp. 431–446, 2007.
- [17] The COIL20 Dataset and COIL100, <http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php>.
- [18] A preprocessed CMU PIE face dataset, http://www.cad.zju.edu.cn/home/dengcai/Data/PIE/results_new.html.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

