Convolutional 2D LDA for Nonlinear Dimensionality Reduction *

Qi Wang[1,2], Zequn Qin[1], Feiping Nie[1], Yuan Yuan[1]
[1]School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL),
Northwestern Polytechnical University, Xian 710072, Shaanxi, P. R. China
[2]Unmanned System Research Institute (USRI),
Northwestern Polytechnical University, Xian 710072, Shaanxi, P. R. China
crabwq@nwpu.edu.cn, qinzequn@mail.nwpu.edu.cn, feipingnie@gmail.com, y.yuan1.ieee@gmail.com

Abstract

Representing high-volume and high-order data is an essential problem, especially in machine learning field. Although existing two-dimensional (2D) discriminant analysis achieves promising performance, the single and linear projection features make it difficult to analyze more complex data. In this paper, we propose a novel convolutional two-dimensional linear discriminant analysis (2D LDA) method for data representation. In order to deal with nonlinear data, a specially designed Convolutional Neural Networks (CNN) is presented, which can be proved having the equivalent objective function with common 2D LDA. In this way, the discriminant ability can benefit from not only the nonlinearity of Convolutional Neural Networks, but also the powerful learning process. Experiment results on several datasets show that the proposed method performs better than other state-of-the-art methods in terms of classification accuracy.

1 Introduction

Linear discriminant analysis (LDA) is a classical method for dimension reduction and classification. It is commonly used in machine learning and pattern recognition, which shows promising performance in applications such as face recognition [Belhumeur *et al.*, 1997]. By maximizing the trace of between-class scatter matrix and minimizing the trace of within-class scatter matrix, the classical LDA aims to find the optimal projection vectors.

The main idea of LDA is simple and effective. The classical LDA, however, demands that input data should be represented by vector. Such constraint is a significant drawback to express complex data. For example, image, the most widely used 2D data format, isn't suitable for vector representation. In order to tackle this problem, one kind of method utilizes the classical LDA combined with matrix-vector transformation [Ma *et al.*, 2013][Ren *et al.*, 2012]. However, what this transformation brings is huge computational expense. In order to process 2D data directly, many method based on 2D data format [Li and Yuan, 2005][Yang *et al.*, 2004] is proposed. Another way to address this problem is two-dimensional linear discriminant analysis (2D LDA) [Ye *et al.*, 2004][Ren *et al.*, 2015], which removes matrix-vector transformation. Using the original 2D-structure data format, 2D LDA can preserve more co-related information which leads to better performance.

With the wave of artificial intelligence, deep version of many classical methods have shown its power again. In recent years, some methods related to classical PCA [Chan *et al.*, 2014] and LDA [Dorfer *et al.*, 2015] achieved promising performance in combination with deep neural networks. The purpose of these methods is to utilize the learning capacity of networks and the effective optimization tool, namely stochastic gradient descent and its variants. However, the combination of classical LDA and deep neural networks means complex network construction and gradient calculation.

In this paper, we propose a convolutional 2D LDA method that aims to solve above the limitation for high-volume and high-order data nonlinear dimensionality reduction. Different from [Dorfer *et al.*, 2015], we employ a special Convolutional Neural Networks (CNN) to optimize LDA objective function instead of maximizing eigenvalues of scatter matrix. The key novelty of our method is that using such CNN structure makes the optimization easier than others and gains better performance. Meanwhile, the whole networks is a nonlinear 2D dimensionality reduction method which optimizes classification and dimensionality reduction networks simultaneously.

2 Related Work

2.1 Revisiting LDA

The classical LDA aims to project the original data into a lower-dimensional space. Meanwhile, the projection should separate the lower-dimensional data. In order to calculate the degree of separation, scatter matrix of projected data is employed. We denote the original data as $X \in R^{l \times n}$, which contains c classes $\pi = [\pi_1, \pi_2, \ldots, \pi_c]$. Suppose the projection is defined by $W \in R^{l \times c}$. The transformation of classical

*This work is supported in part by the National Natural Science Foundation of China under Grant 61379094, in part by the Open Research Fund of Key Laboratory of Spectral Imaging Technology, Chinese Academy of Sciences, and in part by the Fundamental Research Funds for the Central Universities under Grant 3102017AX010. Feiping Nie is the corresponding author.

LDA is $y_i = W^T x_i$, where $x_i \in R^{l\times 1}$ is a sample from the original data. In order to find the optimal projection matrix W, we use between-class scatter matrix S_b and within-class scatter matrix S_w which are defined as follows:

$$S_b = \sum_{i=1}^{c} n_i (M_i - M)(M_i - M)^T,$$

$$S_w = \sum_{i=1}^{c} \sum_{X_j \in \pi_i} (X_j - M_i)(X_j - M_i)^T,$$

where n_i is the number of samples in class π_i, N is the number of whole samples, $M_i = \frac{1}{n_i}\sum_{X_j\in\pi_i} X_j$ is the average value of class π_i and $M = \frac{1}{N}\sum_{i=1}^{c}\sum_{X_j\in\pi_i} X_j$ is the average value of the whole dataset.

Accordingly, the transformed lower-dimensional between-class and within-class scatter matrix can be:

$$\widetilde{S_b} = W^T S_b W,$$
$$\widetilde{S_w} = W^T S_w W.$$

Based on the above condition of separation, the optimal W should be:

$$\max_W \frac{\left\|\widetilde{S_b}\right\|}{\left\|\widetilde{S_w}\right\|}.$$

In [Hou *et al.*, 2012], the above function can be rewritten as follow:

$$\max_W Tr((\widetilde{S_w})^{-1}\widetilde{S_b}),$$

where $Tr(\cdot)$ denotes the matrix trace operation.

2.2 2D LDA

The major differences between LDA and 2D LDA is the data representation format. In order to project 2D-format data, 2D LDA employs a set of transformation matrices. We denotes U, V as the transformation matrices and $X = [X_1, X_2, \ldots, X_n]$ as the input data, where $X_i \in R^{m\times n}$. In this way, the projected between-class and within-class scatter matrix can be rewritten as:

$$\widetilde{S_b} = \sum_{i=1}^{c} n_i U^T (M_i - M) V V^T (M_i - M)^T U,$$

$$\widetilde{S_w} = \sum_{i=1}^{c} \sum_{X_j\in\pi_i} U^T (X_j - M_i) V V^T (X_j - M_i)^T U.$$

Once the scatter matrix of 2D-format data is determined, the same objective function as classical LDA can be formulated. What make difference between objective function of classical LDA and 2D LDA is the transformation matrices. Hence, the optimization target would be a set of transformation matrices U, V:

$$\max_{U,V} \frac{\left\|\widetilde{S_b}\right\|}{\left\|\widetilde{S_w}\right\|}.$$

As mentioned in [Hou *et al.*, 2012], the objective function is defined as follows:

$$\max_{U,V} Tr((\widetilde{S_w})^{-1}\widetilde{S_b}).$$

2.3 Regularized LDA

In the procedure of solving LDA problem, calculating the inverse matrix of within-class scatter matrix is essential. However, the within-class scatter matrix $\widetilde{S_w}$ might be singular under certain circumstances such as tiny sample set of data. This property will make the problem hard to solve. In order to avoid singularity of within-class scatter matrix, regularization terms are added to common LDA problem. Meanwhile, such regularization terms can be helpful to prevent over-fitting problem.

In [Mahanta *et al.*, 2013], an iterative regularized MVLDA is proposed. They use estimates of scatter matrices as regularization terms. The estimates of within-class scatter matrix can be obtained iteratively:

$$S_{WL} = \frac{1}{Nn}\sum_{i=1}^{c}\sum_{X_j\in\pi_i} (X_j - M_i) S_{WR}^{-1} (X_j - M_i)^T,$$

$$S_{WR} = \frac{1}{Nm}\sum_{i=1}^{c}\sum_{X_j\in\pi_i} (X_j - M_i)^T S_{WL}^{-1} (X_j - M_i).$$

The estimates of between-class scatter matrix are defined as:

$$S_{BL} = \sum_{i=1}^{c} n_i (M_i - M)(M_i - M)^T,$$

$$S_{BR} = \frac{1}{Tr(S_{BL})}\sum_{i=1}^{c} n_i (M_i - M)^T (M_i - M).$$

Based on these estimates, the regularized scatter matrices are defined as:

$$S_w^r = (1-\gamma_w) S_w + \gamma_w S_w^s,$$
$$S_b^r = (1-\gamma_b) S_b + \gamma_b S_b^s,$$

where $S_w^s = S_{WR} \otimes S_{WL}$ and $S_b^s = S_{BR} \otimes S_{BL}$.

Another simple regularized LDA utilizes identity matrix to avoid singularity of within-class scatter matrix. The objective function of LDA can be formulated in a unified format:

$$\max Tr((\widetilde{S_w})^{-1}\widetilde{S_b}).$$

If we add an identity matrix to within-class scatter matrix, the whole item will meet the full rank condition. The regularized objective function can be:

$$\max Tr((\widetilde{S_w} + \gamma I)^{-1}\widetilde{S_b}).$$

2.4 Deep Version of LDA

In [Andrew *et al.*, 2013], a Canonical Correlation Analysis (DCCA) method based on deep neural networks is proposed. DCCA shows remarkable results in simultaneously recorded acoustic and articulatory speech data. In [Dorfer *et al.*, 2015], another kind of deep linear discriminant analysis is proposed, which is used for image classification. The original Categorical Cross Entropy is replaced with summation of eigenvalues, which has the same objective function with common 2D LDA.

3 Convolutional 2D LDA

In this section, we expound on the proof of convolutional 2D LDA and it's corresponding CNN construction.

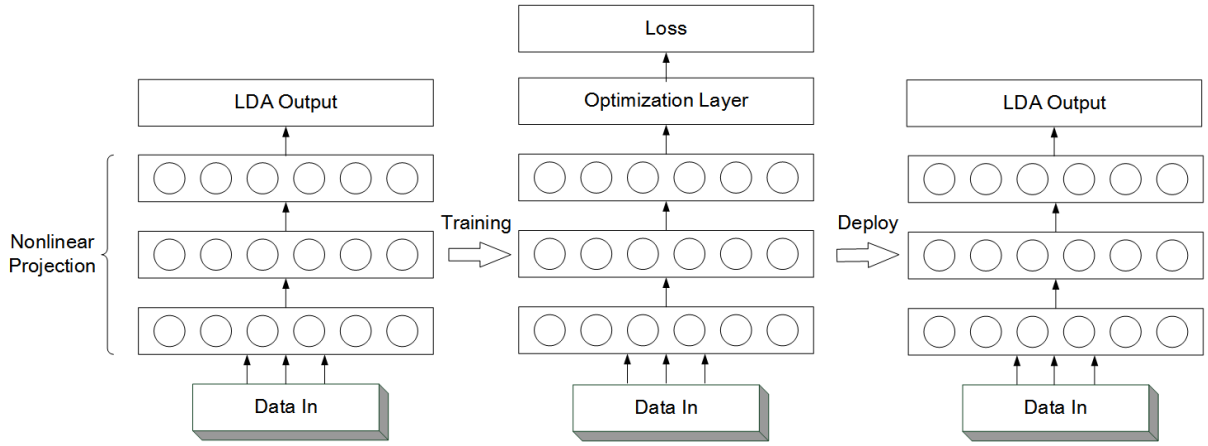


Figure 1: Network construction in different stages. During the training stage, the optimization layer is added to the top of the whole networks. When we deploy the networks, the optimization layer is excluded. The top of whole networks would be the optimized LDA output.

3.1 LDA Based on Nonlinear Projection

Our goal is to find a 2D nonlinear projection that can be used for dimensionality reduction. We denotes $D = [D_1, D_2, \ldots, D_n]$ as the input data and $\psi(\cdot)$ as nonlinear projection. As long as the projected data $d_i = \psi(D_i)$ maximizes the trace of between-class scatter matrix and minimizes the trace of within-class scatter matrix, we can regard such projection as nonlinear 2D LDA dimensionality reduction method.

Different from classical LDA or 2D LDA, what nonlinear projection brings is better representation performance in the projected subspace. However, an essential and difficult barrier is the optimization of nonlinear projection. In order to tackle this problem, we use a special CNN structure to optimize the whole networks. This kind of optimization is the key novelty of our method.

3.2 LDA in Network

Using CNN to realize dimension reduction is our basic idea. Assume that the nonlinear projection defined by CNN is $x = g(A) \in R^{m\times 1}$, in which x is a m-dimensional vector indicating dimension reduction output, $g(\cdot)$ denotes CNN that performs projection and A represents the original data. If the optimal nonlinear projection $g^*(\cdot)$ maximizes the trace of between-class scatter matrix and minimizes the trace of within-class scatter matrix after projection, the optimal nonlinear projection $g^*(\cdot)$ shares the same goal with the classical LDA.

We denote the output of CNN as $X = [g(A_1), g(A_2), \ldots, g(A_n)] \in R^{m\times n}$ and the One-Hot Encoding label of data as $Y \in R^{n\times c}$. Following the realization of regularized LDA, which is helpful to prevent calculation of singular matrix and over-fitting, the objective function is defined as:

$$\max_g Tr((S_t + \gamma I)^{-1} S_b), \tag{1}$$

where

$$S_t = XHX^T, \tag{2}$$

$$S_b = XHY(Y^TY)^{-1}Y^THX^T, \tag{3}$$

$$H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T. \tag{4}$$

Due to the complicated structure of CNN $g(\cdot)$, it is tough to optimize this objective function. Fortunately, we can rewrite the objective function in another way which can be easily optimized:

$$\max_g Tr((S_t + \gamma I)^{-1} S_b)$$

$$\Leftrightarrow \min_{g,W,b} \left\| X^TW + \mathbf{1}b^T - \widetilde{Y} \right\|_F^2 + \gamma \|W\|_F^2, \tag{5}$$

where

$$\widetilde{Y} = Y(Y^TY)^{-\frac{1}{2}}. \tag{6}$$

proof: The equivalence of Eq.1 and Eq.5 can be proved by Lagrange multiplier method. Suppose we have obtained the optimal W and b in Eq.5, the result of substituting them into Eq.5 should be equal to Eq.1. Hence, the optimal solution for b can be obtained by setting the derivative of Eq.5 with respect to b to zero. Thus, we have:

$$b = \frac{1}{n}(\widetilde{Y}^T\mathbf{1} - W^TX\mathbf{1}). \tag{7}$$

Substituting b into Eq.5:

$$\min_{g,W,b} \left\| X^TW + \mathbf{1}b^T - \widetilde{Y} \right\|_F^2 + \gamma \|W\|_F^2$$

$$\Leftrightarrow \min_{g,W} \left\| HX^TW - H\widetilde{Y} \right\|_F^2 + \gamma \|W\|_F^2. \tag{8}$$

Setting the derivative of Eq.8 with respect to W to zeros:

$$W = (XHX^T + \gamma I)^{-1}XH\widetilde{Y}. \tag{9}$$

Substituting W into Eq.8:

$$\min_{g,W} \left\| HX^TW - H\widetilde{Y} \right\|_F^2 + \gamma \|W\|_F^2$$

$$\Leftrightarrow \min_g Tr(\widetilde{Y}^THY) - Tr(\widetilde{Y}^THX^T(XHX^T + \gamma I)^{-1}XH\widetilde{Y}) \tag{10}$$

$$\Leftrightarrow \max_g Tr((S_t + \gamma I)^{-1} S_b).$$

3.3 Network Construction

From the above theorem we can see that if we put a optimization layer above the original CNN $g(\cdot)$ that receives dimension reduction results X and outputs $X^T W + \mathbf{1}b^T$, we would be able to optimize W and b simultaneously.

Meanwhile, the optimal CNN $g(\cdot)$ in Eq.1 can be obtained because Eq.1 shares the same optimal solution with Eq.5 in such circumstances, which means that the dimensionality reduction networks can be optimized via the optimization layer. Furthermore, the new added optimization layer can be used for classification evaluation of dimensionality reduction performance.

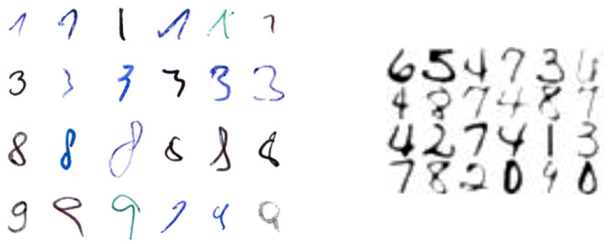
From the 5-layer networks LeNet-5 [Lecun *et al.*, 1998] to ResNet [He *et al.*, 2016] which contains up to 1202 layers, depth of neural network is becoming incredibly huge. The purpose of our method, however, is to examine the effectiveness of this kind of specially designed networks instead of using the great generalization ability of deep networks. Thus, we use two kinds of network structure to examine our method. A simplified dimensionality reduction networks is employed that contains two convolutional layers and one fully connected layer to compare with traditional algorithms. A more complex networks which contains three convolutional layers with batch normalization [Ioffe and Szegedy, 2015] and one fully connected layer with dropout[Hinton *et al.*, 2012] is used to compare with other algorithms based on deep learning.

As mentioned in Eq.5, we use one single layer to perform classification above the dimensionality reduction networks. It converts the dimensionality reduction results into One-Hot encoding label in class space, which can be regarded as classification networks. Meanwhile, this classification networks is in charge of optimizing the dimensionality reduction networks. The loss of whole networks is defined as the Frobenius norm of difference between classification and label, as shown in Eq.5. Figure 1 illustrates the training and deploy procedure. In order to optimize the whole networks, the extra optimization layer is added in the training stage. It can be used as classifier and optimizer simultaneously. Once the optimization of whole networks is finished, the optimization layer can be excluded from original networks.

In this way, we can combine dimensionality reduction and classification stage into a end-to-end networks, which shares the same objective function with regularized 2D LDA. Apparently, our method is a nonlinear 2D LDA method.

4 Experiment

In this section, we compare the proposed convolutional 2D LDA with eight traditional algorithms, including LDA [Belhumeur *et al.*, 1997], 2D LDA [Ye *et al.*, 2004], 2D PCA [Yang *et al.*, 2004], Bilinear SVM [Pirsiavash *et al.*, 2009], S2D LDA [Inoue and Urahama, 2006], P2D LDA [Inoue and Urahama, 2006], Tensor LPP [He *et al.*, 2005] and CRP [Chang *et al.*, 2015]. Experiments are performed on two handwritten digit datasets. We also compare our method with other four algorithms based on deep learning, including NIN [Lin *et al.*, 2013], Maxout [Goodfellow *et al.*, 2013], DeepCNet [Graham, 2014] and DeepLDA [Dorfer *et al.*, 2015]. In addition, we provide detailed networks architecture and hyper parameters settings used in our experiment.



(a) (b)

Figure 2: Example images in (a) CVL dataset and (b) USPS dataset.

4.1 Datasets and Experiment Setup

We use three datasets to conduct our experiments.

MNIST dataset: The MNIST dataset contains 60,000 examples used for handwritten digits recognition.

CVL dataset: The CVL dataset [Diem *et al.*, 2013] is generated for the ICDAR2013 Handwritten Digit Recognition Competition. There are 21,780 handwritten digit images in this dataset. The size of each image is 32×32.

USPS dataset: The USPS dataset is used for light weight handwritten digit recognition. It contains 9,298 handwritten digit images in this database with the size of 16×16.

As mentioned in section 3.2, we adopt two kinds of networks with two and three convolutional layers, one fully connected layer and it's corresponding LDA classification and optimization layer. In Table 1 we outline the architecture of our model in detail. The whole networks is trained with momentum [Sutskever *et al.*, 2013] and Adam [Kingma and Ba, 2014] SGD optimizer in TensorFlow[Abadi *et al.*, 2016]. For a more convincing result, all hyper parameters for different datasets are identical. It is noteworthy that batch size is an essential hyper parameter because lack of certain classes in a mini-batch would make the weighted label $\widetilde{Y}$ meaningless. The batch size is set to 100 and the learning rate to 0.0002. The regularization weight γ is set to 0.0001.

Table 1: The architecture of the Convolutional 2D LDA Network.

stage	simplified	complex
conv1	3x3,32conv-RELU 2x2 Max-Pooling	3x3,64conv-BN-RELU 3x3,64conv-BN-RELU 2x2 Max-Polling
conv2	3x3,64conv-RELU 2x2 Max Pooling	3x3,96conv-BN-RELU 3x3,96conv-BN-RELU 2x2 Max-Polling
conv3		3x3,256conv-BN-RELU 1x1,256conv-BN-RELU 1x1,64conv-BN-RELU
fc1	64d	64d-Dropout(0.5)
optimization	10d	10d

Table 2: Classification accuracy of traditional algorithms and our method using 80% training data for each dataset.

Dataset	LDA	2DPCA	2DLDA	S2DLDA	P2DLDA	T-LPP	B-SVM	CRP	Our method (simplified)
CVL(1NN)	90.2±1.6	90.6±1.5	91.1±1.4	92.4±1.5	92.8±1.3	93.1±1.3	93.4±1.5	94.7±1.3	**96.6**
CVL(SVM)	87.3±1.5	87.7±1.7	88.2±1.4	89.3±1.1	89.9±1.1	90.3±1.5	93.4±1.5	91.5±1.3	**96.6**
USPS(1NN)	94.8±1.3	95.2±1.1	95.6±1.2	96.4±1.4	94.5±1.2	95.1±1.1	96.2±1.4	96.8±0.9	**97.9**
USPS(SVM)	93.1±1.6	93.5±1.5	93.8±1.1	94.1±1.2	94.3±1.8	94.7±1.4	96.2±1.4	95.6±1.3	**97.9**

Table 3: Classification accuracy of traditional algorithms and our method using 20 training data for each dataset.

Dataset	LDA	2DPCA	2DLDA	S2DLDA	P2DLDA	T-LPP	B-SVM	CRP	Our method (simplified)
CVL(1NN)	63.7±1.3	64.1±1.8	66.7±1.2	67.3±1.3	60.2±1.7	65.2±1.9	70.9±1.6	**74.2±1.1**	72.1
CVL(SVM)	67.9±1.3	68.3±1.4	69.2±1.6	68.6±1.7	58.3±1.7	69.1±1.4	70.9±1.6	**79.2±1.5**	72.1
USPS(1NN)	83.1±1.3	83.6±1.4	84.5±1.8	85.6±1.2	74.7±1.5	79.5±1.4	86.6±1.8	89.2±1.4	**90.2**
USPS(SVM)	83.8±1.8	84.2±1.9	85.8±1.5	86.8±1.9	80.8±1.3	81.8±1.6	86.6±1.8	88.4±1.4	**90.2**

4.2 Experimental Results

Traditional Method

In this experiment, we use classification accuracy to compare our algorithm with other eight methods, including LDA, 2D LDA, 2D PCA, Bilinear SVM, S2D LDA, P2D LDA, Tensor LPP and CRP. Two different classifiers are used in this experiment which are SVM and 1-Nearest-Neighbor (1NN). Note that our method utilizes the extra optimization layer to classify, instead of SVM or 1NN. Because the Bilinear SVM itself is a classifier, we compare it with our method directly. In order to evaluate the effectiveness of our method, three different sizes of training sets for both datasets are employed.

The experiment results using 80% training data for each dataset are shown in Table 2. Our method outperforms other eight methods in different datasets and classifiers. This result shows that our method takes the advantages of deep neural networks. Meanwhile, the effectiveness of our method can be verified. When we use CVL dataset with SVM classifier, our method outperforms classical 2D LDA by 8.9% in terms of classification accuracy.

The experiment results using 20 training data for each dataset are shown in Table 3. The CRP method achieves the best results in CVL dataset using SVM and 1NN classifier. The second best one is our method. In USPS dataset, our method consistently performs better than the other method. What we can see from this experiment is that the CNN starts to suffer from Insufficient data.

The experiment results using 10 training data for each dataset are shown in Table 5. With the decrease of training data, all the methods show worse results. Our method achieves the 3rd place in CVL dataset using 1NN classifier. When it comes to SVM classifier, our method shows barely satisfactory results. In USPS dataset, our method achieves the 2nd place compared with other algorithms. Although the number of training data is extremely small, our method still outperforms 2D LDA by 10.8% in USPS dataset using 1NN as classifier.

Deep Learning Method

In this experiment, we use classification accuracy to compare our algorithm with several deep learning method in MNIST dataset, including NIN, Maxout, DeepCNet, DeepLDA. Note that we use the simplified and the complex model to compare. In Table 4, we can observe that our method achieves promising results. Although our simplified version method shows poor performance compared with other method, we can still say that accuracy of 99.2% is not a bad result in consideration of its structure.

Table 4: Classification accuracy of deep learning algorithms.

Method	Classification accuracy
NIN	99.53
Maxout	99.55
DeepCNet	**99.69**
DeepLDACCE	99.66
DeepLDA	99.68
Our method(simplified)	99.20
Our method(complex)	**99.69**

From the above experiments, we can observe that the proposed method performs fairly good in rich data context. Compared with classical 2D LDA, our method achieves better performance in all of the datasets and classifiers. when dealing with insufficient data, the proposed method still shows a satisfactory result. Taking into account all these three experi-

Table 5: Classification accuracy of traditional algorithms and our method using 10 training data for each dataset.

Dataset	LDA	2DPCA	2DLDA	S2DLDA	P2DLDA	T-LPP	B-SVM	CRP	Our method (simplified)
CVL(1NN)	47.3±1.4	47.9±1.5	50.3±1.8	51.9±1.5	46.9±1.4	55.1±1.6	64.2±1.9	**67.3±1.3**	55.2
CVL(SVM)	56.8±1.5	57.1±1.6	57.9±1.9	51.9±1.4	48.4±1.3	66.9±1.6	64.2±1.9	**68.3±1.5**	55.2
USPS(1NN)	68.5±1.7	69.2±1.5	71.8±1.4	71.2±1.6	64.8±1.9	76.7±1.6	79.4±1.9	**84.4±1.5**	82.6
USPS(SVM)	73.4±1.7	74.4±1.9	77.3±1.3	79.2±1.8	73.9±1.4	78.2±1.5	79.4±1.9	**84.3±1.3**	82.6

ments, we can say that our method achieves a promising performance.

4.3 Discussion

It is interesting to see that our method gets a more satisfying performance in rich data context compared with insufficient data condition. This drawback might be caused by three reasons.

First, according to Eq.5, the optimal dimensionality reduction networks can be obtained with optimal W and b. With insufficient data, the final dimensionality reduction networks is optimized with non-optimal W and b.

Second, the over-fitting problem in deep neural networks would be serious when we use small dataset. In Table 5, the whole training dataset contains only 100 samples while deep neural networks needs a large number of data.

The last reason is the training trick. All of the experiments are conducted with the same hyper parameters. If the hyper parameters are adjusted for insufficient data, the classification accuracy would be higher. In Table 6 and Table 7, the hyper parameters adjusted for insufficient data lead to a better performance. All we have done is setting the learning rate higher to overcome the over-fitting problem. The our method$^+$ in Table 6 and Table 7 uses learning rate of 0.0004 and 0.0006 respectively. The learning rate used in our original experiment is 0.0002.

Table 6: Classification accuracy with different hyper parameter using 20 training data. Best method means best of other eight traditional algorithms.

Dataset	Best	Our method (simplified)	Our method $^+$ (simplified)
CVL(1NN)	74.2 ± 1.1	72.1	**75.3**
CVL(SVM)	**79.2 ± 1.5**	72.1	75.3
USPS(1NN)	89.2 ± 1.4	90.2	**91.4**
USPS(SVM)	88.4± 1.4	90.2	**91.4**

When we increase the number of training data from 100 to 200 in CVL dataset, the average classification accuracy improvement of our method and CRP are 16.9% and 8.9%. When we use more data, the average classification accuracy improvement of our method and CRP are 24.5% and 16.4%. Such phenomenon can be seen in USPS dataset as well. Compared with CRP algorithm that shows great performance using few data, our method shows much faster performance growth with the increasement of training data number.

The experiment results show two major features of our method. One of them is that our method need more data than other methods. Another one is that better performance can be achieved with sufficient data. As a result of CNN structure, this kind of features are obvious.

Table 7: Classification accuracy with different hyper parameter using 10 training data. Best method means best of other eight traditional algorithms.

Dataset	Best	Our method (simplified)	Our method $^+$ (simplified)
CVL(1NN)	**67.3 ± 1.3**	55.2	62.9
CVL(SVM)	**68.3 ± 1.5**	55.2	62.9
USPS(1NN)	84.4 ± 1.5	82.6	**86.2**
USPS(SVM)	84.3± 1.3	82.6	**86.2**

5 Conclusion

In this paper, we have proposed a convolutional 2D LDA method for nonlinear dimensionality reduction. The difficult problem of optimization is solved by a clever equivalence of two objective functions. The proposed method employs a two stage end-to-end CNN to realize dimensionality reduction. Effectiveness of such structure has been proved with two different networks. Our convolutional 2D LDA method outperforms the classical LDA in all experiment settings. With sufficient data, our method shows remarkable dimensionality reduction ability.

References

[Abadi *et al.*, 2016] Martn Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard. Tensorflow: A system for large-scale machine learning. 2016.

[Andrew *et al.*, 2013] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, pages 1247–1255, 2013.

[Belhumeur *et al.*, 1997] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[Chan *et al.*, 2014] Tsung Han Chan, Kui Jia, Shenghua Gao, and Jiwen Lu. Pcanet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, 24(12):5017–5032, 2014.

[Chang *et al.*, 2015] Xiaojun Chang, Feiping Nie, Sen Wang, Yi Yang, Xiaofang Zhou, and Chengqi Zhang. Compound rank-k projections for bilinear analysis. 2015.

[Diem *et al.*, 2013] Markus Diem, Stefan Fiel, Angelika Garz, Manuel Keglevic, Florian Kleber, and Robert Sablatnig. Icdar 2013 competition on handwritten digit recognition (hdrc 2013). pages 1422–1427, 2013.

[Dorfer *et al.*, 2015] Matthias Dorfer, Rainer Kelz, and Gerhard Widmer. Deep linear discriminant analysis. *arXiv preprint arXiv:1511.04707*, 2015.

[Goodfellow *et al.*, 2013] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C Courville, and Yoshua Bengio. Maxout networks. *ICML (3)*, 28:1319–1327, 2013.

[Graham, 2014] Benjamin Graham. Spatially-sparse convolutional neural networks. *arXiv preprint arXiv:1409.6070*, 2014.

[He *et al.*, 2005] Xiaofei He, Deng Cai, and Partha Niyogi. Tensor subspace analysis. In *Advances in neural information processing systems*, pages 499–506, 2005.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[Hinton *et al.*, 2012] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *Computer Science*, 3(4):pgs. 212–223, 2012.

[Hou *et al.*, 2012] Y. Hou, I. Song, H. K. Min, and C. H. Park. Complexity-reduced scheme for feature extraction with linear discriminant analysis. *IEEE Transactions on Neural Networks & Learning Systems*, 23(6):1003–1009, 2012.

[Inoue and Urahama, 2006] Kohei Inoue and Kiichi Urahama. Non-iterative two-dimensional linear discriminant analysis. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 540–543. IEEE, 2006.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Computer Science*, 2015.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Lecun *et al.*, 1998] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li and Yuan, 2005] Ming Li and Baozong Yuan. 2d-lda: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters*, 26(5):527–532, 2005.

[Lin *et al.*, 2013] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[Ma *et al.*, 2013] Zhigang Ma, Yi Yang, Feiping Nie, and Nicu Sebe. Thinking of images as what they are: compound matrix regression for image classification. In *International Joint Conference on Artificial Intelligence*, pages 1530–1536, 2013.

[Mahanta *et al.*, 2013] Mohammad Shahin Mahanta, Amirhossein S. Aghaei, and Konstantinos N. Plataniotis. Regularized lda based on separable scatter matrices for classification of spatio-spectral eeg patterns. 32(3):1237–1241, 2013.

[Pirsiavash *et al.*, 2009] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Bilinear classifiers for visual recognition. In *Advances in neural information processing systems*, pages 1482–1490, 2009.

[Ren *et al.*, 2012] Chuan-Xian Ren, Dao-Qing Dai, and Hong Yan. Coupled kernel embedding for low-resolution face image recognition. *IEEE Transactions on Image Processing*, 21(8):3770–3783, 2012.

[Ren *et al.*, 2015] Chuan-Xian Ren, DAI Dao-Qing, Xiaofei He, and Hong Yan. Sample weighting: An inherent approach for outlier suppressing discriminant analysis. *IEEE Transactions on Knowledge and Data Engineering*, 27(11):3070–3083, 2015.

[Sutskever *et al.*, 2013] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, 2013.

[Yang *et al.*, 2004] Jian Yang, David Zhang, Alejandro F Frangi, and Jing-yu Yang. Two-dimensional pca: a new approach to appearance-based face representation and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 26(1):131–137, 2004.

[Ye *et al.*, 2004] Jieping Ye, Ravi Janardan, and Qi Li. Two-dimensional linear discriminant analysis. In *Advances in neural information processing systems*, pages 1569–1576, 2004.