Neural Process Lett (2013) 38:17–27
DOI 10.1007/s11063-012-9260-y

Convolutional Deep Networks for Visual Data Classification

Shusen Zhou · Qingcai Chen · Xiaolong Wang

Published online: 20 November 2012

Abstract This paper develops a semi-supervised learning algorithm called convolutional deep networks (CDN), to address the image classification problem with deep learning. First, we construct the previous several hidden layers using convolutional restricted Boltzmann machines, which can reduce the dimension and abstract the information of the images effectively. Second, we construct the following hidden layers using restricted Boltzmann machines, which can abstract the information of images quickly. Third, the constructed deep architecture is fine-tuned by gradient-descent based supervised learning with an exponential loss function. CDN can reduce the dimension and abstract the information of the images at the same time efficiently. More importantly, the abstraction and classification procedure of CDN use the same deep architecture to optimize the same parameter in different steps continuously, which can improve the learning ability effectively. We did several experiments on two standard image datasets, and show that CDN are competitive with both representative semi-supervised classifiers and existing deep learning techniques.

Keywords Semi-supervised learning · Deep learning · Convolutional neural networks · Visual data classification

1 Introduction

Recently, more and more people use digital photography technology, huge image collections are available through the Internet, which create a need for image processing [14]. For the wide

S. Zhou (✉) · Q. Chen · X. Wang
Key Laboratory of Network Oriented Intelligent Computation, Shenzhen Graduate School, Harbin Institute of Technology, Harbin, People's Republic of China
e-mail: zhoushusen@gmail.com

Q. Chen
e-mail: qingcai.chen@hitsz.edu.cn

X. Wang
e-mail: wangxl@insun.hit.edu.cn

application prospect, much effort has been directed towards recognizing objects and classifying images [4]. However, in real-world applications, it is often the case that the labeled data are difficult, expensive, or time consuming to obtain [1], while abundant of unlabeled data are available. To address this problem, semi-supervised learning, which uses large amount of unlabeled data together with labeled data to build better learners, has attracted more and more attention [17].

Most semi-supervised methods use shallow architecture to model the problem [16]. Recently, several methods have been proposed based on deep architecture, which is expected to perform well in semi-supervised learning. Weston et al. leverage shallow algorithms to deep architectures and yield competitive performance in semi-supervised learning task [16]. Deep belief networks (DBN) is a representative deep learning algorithm, which is a directed belief nets with many hidden layers constructed by restricted Boltzmann machines (RBM), and refined by a gradient-descent based supervised learning [7,8]. The two-stage construction of DBN makes it natural to semi-supervised learning. DBN-rNCA [15], which combines the DBN and neighborhood component analysis (NCA) techniques, also demonstrates the good performance for classification task via semi-supervised learning. Liu et al. propose a novel semi-supervised classifier called discriminative deep belief networks (DDBN), which utilizes a new deep architecture to integrate the abstraction ability of DBN and discriminative ability of backpropagation strategy [13].

Convolutional neural networks (CNN) are specifically designed to deal with the variability of two dimensional shapes, represent one of the early successes of deep learning [9]. Lecun et al. compare various handwritten character recognition methods on a standard handwritten digit recognition task, and shown CNN outperforms all other techniques [10]. Desjardins and Bengio adapt RBM to operate in a convolutional manner, and show that the convolutional restricted Boltzmann machines (CRBM) are more efficient than standard RBM [3]. Lee et al. present the convolutional deep belief network (CDBN), a hierarchical generative model scales to realistic image sizes [11]. CDBN is constructed by CRBM layer by layer, and the representation is subsampled by probabilistic max-pooling in every layer, which can extract the features of images with unsupervised learning effectively, then the features can be classified by support vector machines (SVM).

In this paper, we propose a semi-supervised classifier called convolutional deep networks (CDN) based on two representative deep architecture CNN and DBN. CDN is constructed by greedy layer-wise unsupervised learning, the bottom layers are constructed by CRBM, and the upper layers are constructed by RBM, then the whole constructed deep architecture is fine tuned by a gradient-descent based supervised learning based on an exponential loss function.

The remainder of this paper is organized as follows. In Sect. 2, we introduce our semi-supervised learning method CDN in details. Section 3 shows the empirical validation of CDN by comparing its classification performance with previous semi-supervised learning methods and deep learning methods on image datasets. The paper is closed with conclusion.

2 Convolutional Deep Networks

In this part, we propose a semi-supervised learning algorithm, convolutional deep networks (CDN) based on the representative deep architecture CNN and DBN for image classification. We formulate the problem in Sect. 2.1 and provide the solution via deep architecture in Sect. 2.2. Section 2.3 discusses greedy layer-wise unsupervised learning with CRBM and RBM

separately. Section 2.4 provides gradient descent supervised learning with an exponential loss function. Section 2.5 introduces training and test procedure of CDN.

2.1 Problem Formulation

Let $\mathbf{X}$ be a set of images, each image composed of $D \times D$ pixels, which can be written as:

$$\mathbf{X} = \left[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{L+T}\right] \tag{1}$$

where $\mathbf{x}$ is a $D \times D$ pixels image. L is the number of labeled images, T is the number of unlabeled images.

Let $\mathbf{Y}$ be a set of labels correspond to L labeled images and is denoted as:

$$\mathbf{Y}^L = \left[\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^L\right] \tag{2}$$

where $\mathbf{y}$ is a vector with C units, C is the number of classes.

$$y_j = \begin{cases} 1 & \text{if} \quad \mathbf{x} \in j^{\text{th}} \text{ class} \\ -1 & \text{if} \quad \mathbf{x} \notin j^{\text{th}} \text{ class} \end{cases} \tag{3}$$

We intend to seek the mapping function $\mathbf{X} \rightarrow \mathbf{Y}$ using the L labeled data and T unlabeled data. After training, we can determine $\mathbf{y}$ using the mapping function when a new sample $\mathbf{x}$ comes.

2.2 Architecture of CDN

The architecture of CDN can be seen in Fig. 1, which is a fully interconnected directed belief nets with one input layer $\mathbf{h}^0$, N hidden layers $\mathbf{h}^1, \mathbf{h}^2, \ldots, \mathbf{h}^N$, and one label layer at the top. The input layer $\mathbf{h}^0$ has $D \times D$ units, equal to the number of pixels of sample image $\mathbf{x}$. The hidden layer has M layers constructed by 2 dimensional CRBM and $N - M$ layers constructed by RBM. The label layer has C units, equal to the number of classes. The seeking of the mapping function $\mathbf{X} \rightarrow \mathbf{Y}$, here, is transformed to the problem of finding the parameter space $\mathbf{W} = \{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^N\}$ for the deep architecture.

The training of the CDN can be divided into two stages:

1. CDN is constructed by greedy layer-wise unsupervised learning using CRBM and RBM as building blocks.
2. CDN is optimized by an exponential loss function through back propagation supervised learning.

2.3 Unsupervised Learning

As shown in Fig. 1, we construct CDN layer by layer using CRBM and RBM. The architecture of CRBM can be seen in Fig. 2, which is a two-layer recurrent neural network in which stochastic inputs groups are connected to stochastic outputs groups using symmetrically weighted connections. The top layer represents a vector of stochastic hidden feature $\mathbf{h}^k$ and the bottom layer represents a vector of visible data $\mathbf{h}^{k-1}$, $k = 1, \ldots, M$. The kth layer consists of G_k groups, each group consists of $D_k \times D_k$ units. The dimension of each group is reduced through subsampling with mean method, after subsampling, each group consists of $\hat{D}_k \times \hat{D}_k$ units, resulting in $G_k \times \hat{D}_k \times \hat{D}_k$ hidden units. The input layer $\mathbf{h}^0$ is consists

Fig. 1 Architecture of CDN

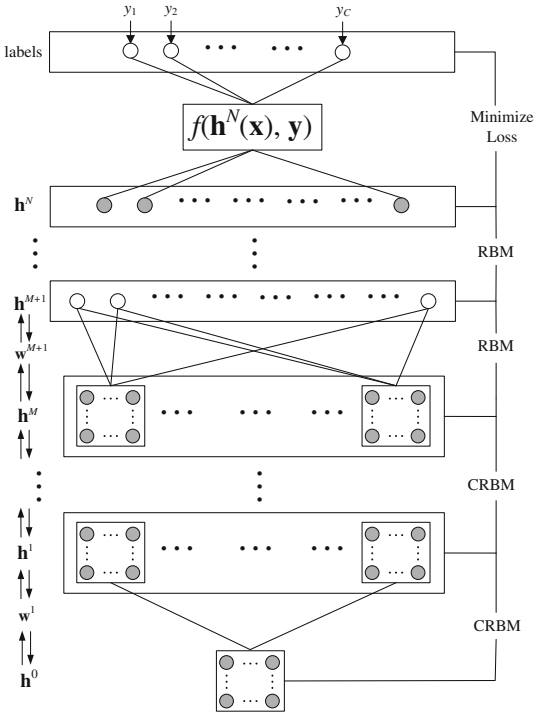
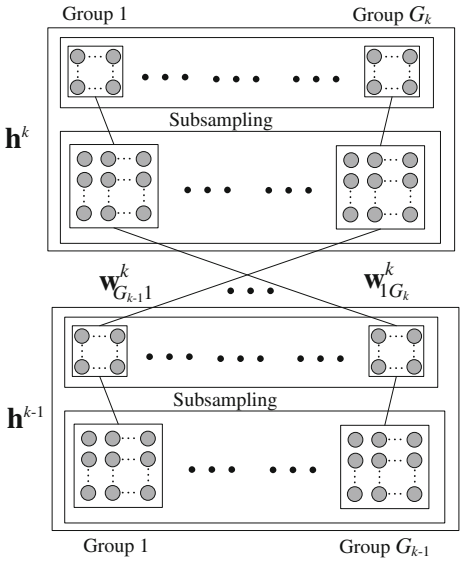


Fig. 2 Architecture of CRBM



of one group and $D \times D$ units, which represent one input image. $\mathbf{w}^k$ is the symmetric interaction term connecting corresponding groups between data $\mathbf{h}^{k-1}$ and feature $\mathbf{h}^k$. However, the weights of CRBM between the hidden and visible groups are shared among all locations [11], and the calculation is operated in a convolutional manner [3].

We define the energy of the state $(\mathbf{h}^{k-1}, \mathbf{h}^k)$ as:

$$E\left(\mathbf{h}^{k-1}, \mathbf{h}^k; \theta\right) = -\sum_{s=1}^{G_{k-1}} \sum_{t=1}^{G_k} (\tilde{w}_{st}^k * h_s^{k-1}) \cdot h_t^k - \sum_{s=1}^{G_{k-1}} b_s^{k-1} \sum_{u=1}^{\hat{D}_{k-1}} h_s^{k-1} - \sum_{t=1}^{G_k} c_t^k \sum_{v=1}^{D_k} h_t^k \tag{4}$$

where $\theta = (\mathbf{w}, \mathbf{b}, \mathbf{c})$ are the model parameters: w_{st}^k is a filter between unit s in the layer $\mathbf{h}^{k-1}$ and unit t in the layer $\mathbf{h}^k$, $k = 1, \ldots, M$. The dimension of the filter w_{st}^k is equal to $(\hat{D}_{k-1} - D_k + 1) \times (\hat{D}_{k-1} - D_k + 1)$. b_s^{k-1} is the sth bias of layer $\mathbf{h}^{k-1}$ and c_t^k is the tth bias of layer $\mathbf{h}^k$. A tilde above an array $(\tilde{w})$ denote flipping the array, $*$ denote valid convolution, and $\cdot$ denote element-wise product followed by summation, i.e., $A \cdot B = trA^T B$ [11].

The joint and conditional probability distribution are defined as follows:

$$P\left(\mathbf{h}^{k-1}; \theta\right) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}^k} \exp\left(-E\left(\mathbf{h}^{k-1}, \mathbf{h}^k; \theta\right)\right) \tag{5}$$

$$Z(\theta) = \sum_{\mathbf{h}^{k-1}} \sum_{\mathbf{h}^k} \exp\left(-E\left(\mathbf{h}^{k-1}, \mathbf{h}^k; \theta\right)\right) \tag{6}$$

where $Z(\theta)$ denotes the normalizing constant.

The probability of turning on unit v in group t is a logistic function of the states of $\mathbf{h}^{k-1}$ and w_{st}^k:

$$p\left(h_{t,v}^k = 1 | \mathbf{h}^{k-1}\right) = \text{sigm}\left(c_t^k + \left(\sum_s \tilde{w}_{st}^k * h_s^{k-1}\right)_v\right) \tag{7}$$

The probability of turning on unit u in group s is a logistic function of the states of $\mathbf{h}^k$ and w_{st}^k:

$$p\left(h_{s,u}^{k-1} = 1 | \mathbf{h}^k\right) = \text{sigm}\left(b_s^{k-1} + \left(\sum_t w_{st}^k \star h_t^k\right)_u\right) \tag{8}$$

where the logistic function is:

$$\text{sigm}(\eta) = 1/(1 + e^{-\eta}) \tag{9}$$

A star $\star$ denote full convolution.

The derivative of the log-likelihood with respect to the model parameter $\mathbf{w}^k$ can be obtained by the CD method [5,6]:

$$\frac{\partial \log p(\mathbf{h}^{k-1})}{\partial w_{st}^k} = \left\langle h_s^{k-1} h_t^k \right\rangle_{P_0} - \left\langle h_s^{k-1} h_t^k \right\rangle_{P_M} \tag{10}$$

where $\langle \cdot \rangle_{P_0}$ denotes an expectation with respect to the data distribution and $\langle \cdot \rangle_{P_M}$ denotes a distribution of samples from running the Gibbs sampler initialized at the data, for M full steps.

The detail introduction of RBM can be seen in Hinton et al. [8].

2.4 Supervised Learning

In CDN, we construct the deep architecture using all labeled images with unlabeled images by inputting them one by one from layer $\mathbf{h}^0$. The deep architecture is constructed layer by layer from bottom to top, and each time, the parameter $\mathbf{w}^k$ is trained by the calculated data in the $k-1$th layer.

According to the $\mathbf{w}^k$ calculated by CRBM and RBM, the layer $\mathbf{h}^k, k = 1, \ldots, M$ can be got as following when a sample $\mathbf{x}$ inputs from layer $\mathbf{h}^0$:

$$h_t^k(\mathbf{x}) = \text{sigm}\left(c_t^k + \sum_{s=1}^{G_{k-1}} \tilde{w}_{st}^k * h_s^{k-1}(\mathbf{x})\right), t = 1, \ldots, G_k \tag{11}$$

Then CDN subsampling $h_t^k(\mathbf{x})$ with mean method in every nonoverlapping 2×2 neighborhood, which result in half the number of rows and columns for every group of $h^k(\mathbf{x})$.

When $k = M+1, \ldots, N-1$, the layer $\mathbf{h}^k$ can be represented as:

$$h_t^k(\mathbf{x}) = \text{sigm}\left(c_t^k + \sum_{s=1}^{D_{k-1}} w_{st}^k h_s^{k-1}(\mathbf{x})\right), t = 1, \ldots, D_k \tag{12}$$

The parameter $\mathbf{w}^N$ is initialized randomly, just as backpropagation algorithm.

$$h_t^N(\mathbf{x}) = c_t^N + \sum_{s=1}^{G_{N-1} \times D_{N-1}} w_{st}^N h_s^{N-1}(\mathbf{x}), \ t = 1, \ldots, D_N \tag{13}$$

After greedy layer-wise unsupervised learning, $\mathbf{h}^N(\mathbf{x})$ is the representation of $\mathbf{x}$. Then we use L labeled images to refine the parameter space $\mathbf{W} = \{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^N\}$ for better discriminative ability. This task can be formulated as an optimization problem:

$$\underset{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^N}{\arg\min} \quad \text{f}\left(h^N\left(\mathbf{X}^L\right), \mathbf{Y}^L\right) \tag{14}$$

where

$$\text{f}\left(h^N\left(\mathbf{X}^L\right), \mathbf{Y}^L\right) = \sum_{i=1}^{L} \sum_{j=1}^{C} \text{T}\left(h_j^N\left(\mathbf{x}^i\right) y_j^i\right) \tag{15}$$

and the loss function is defined as

$$T(r) = \exp(-r) \tag{16}$$

We use gradient-descent through the whole CDN to refine the weight space. In the supervised learning stage, the stochastic activities are replaced by deterministic, real valued probabilities.

2.5 Classification Using CDN

The training procedure of CDN is given in Algorithm 1. For the training of CDN architecture, the parameters are random initialized with normal distribution. All the reviews in the dataset are used to train the CDN with unsupervised learning. The number of units $D_1, \ldots, D_N$ in hidden layer, the number of groups $G_M, \ldots, G_N$ in convolutional layer, and the number of epochs Q are set manually based on the dimension of the input data and the size of dataset.

Algorithm 1: Algorithm of CDN

Input: data $\mathbf{X}$, $\mathbf{Y}^L$
number of units in every hidden layer $D_1 \dots D_N$
number of groups in every convolutional hidden layer $G_M \dots G_N$
number of layers N; number of epochs Q; number of iterations I
number of labeled data L; number of unlabeled data T
hidden layer $\mathbf{h}^1, \dots, \mathbf{h}^M$; convolutional hidden layer $\mathbf{h}^{M+1}, \dots, \mathbf{h}^{N-1}$
parameter space $\mathbf{W} = \{\mathbf{w}^1, \dots, \mathbf{w}^N\}$; biases $\mathbf{b}$, $\mathbf{c}$; momentum ϑ and learning rate η
Output: deep architecture with parameter space $\mathbf{W}$
for $i = 1; i \le I$ **do**
  1. Greedy layer-wise unsupervised learning
  for $k = 1; k \le N - 1$ **do**
    for $q = 1; q \le Q$ **do**
      for $r = 1; r \le L + T$ **do**
        Calculate the non-linear positive and negative phase:
        if $k \le M$ **then**
          Normal calculation according to RBM.
        else
          Convolutional calculation according to Eq. 7 and Eq. 8.
        end
        Update the weights and biases:
        $$w_{st}^k = \vartheta\, w_{st}^k + \eta \left(\left\langle h_{s,r}^{k-1} h_{t,r}^k \right\rangle_{P_0} - \left\langle h_{s,r}^{k-1} h_{t,r}^k \right\rangle_{P_1} \right)$$
      end
    end
  end
  2. Supervised learning based on gradient descent
  $$\arg\min_{W} \sum_{i=1}^{L} \sum_{j=1}^{C} \exp(-h^N(x_j^i) y_j^i)$$
end

After training, we can use the Eq. 17 to determine the label of the new data.

$$\arg\max_j h^N(\mathbf{x}) \tag{17}$$

where j is the coordinate of the vector $h^N(\mathbf{x})$ with maximum value.

3 Experiments

3.1 Experimental Setup

We evaluate the performance of the proposed CDN method using two image classification datasets. The first dataset is Caltech 101, a standard dataset for image classification, including the images of 101 different objects, plus a background category [12]. The second dataset is MNIST, a standard dataset for empirical validation of deep learning algorithms [15,16].

We compare the classification performance of CDN with four representative classifiers, Transductive SVM (TSVM) [2], EmbedNN [16], DBN-rNCA [15], and DDBN. TSVM is the semi-supervised version of SVM, EmbedNN is the semi-supervised version of NN with deep architecture, and DBN-rNCA is the semi-supervised version of DBN. DDBN is a semi-supervised learning method based on DBN proposed recently [13]. We repeat all experiments

20 times on randomly selected labeled and unlabeled images, and report average and variance of error rates.

3.2 Caltech 101 Dataset

We work on the subset of Caltech 101, which includes 2,935 images from the first five categories. We preprocess the images to the same size, every image include 20×20 pixels. The CDN structure used in this experiment is 8-50-200-5, which represents the number of groups in two convolutional hidden layers are 8 and 50, the number of units in one normal hidden layer and output layer are 200 and 5 respectively.

We set the number of labeled images equal to 25, 50, 75, and compare the classification error rate of different methods with these various numbers of labeled images respectively. As shown in Figs. 3, 4 and 5, the performance of CDN is better than other classifiers. Although

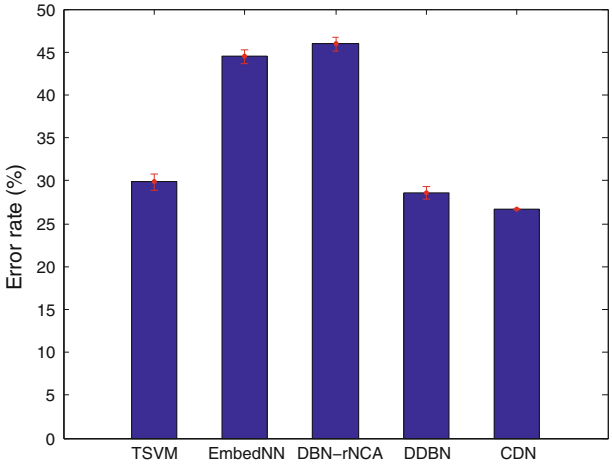


Fig. 3 Average and variance of error rates with 25 labeled data on Caltech 101

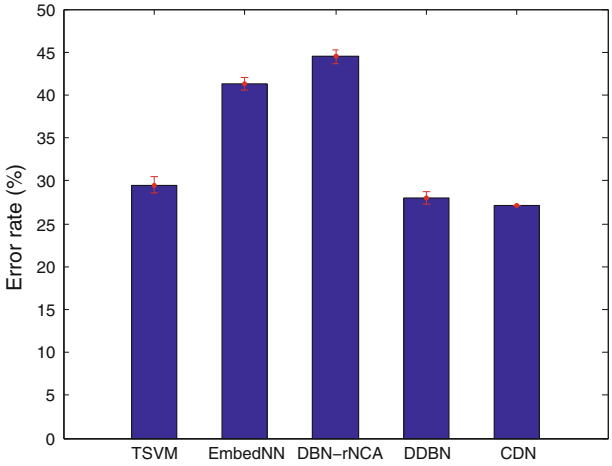


Fig. 4 Average and variance of error rates with 50 labeled data on Caltech 101

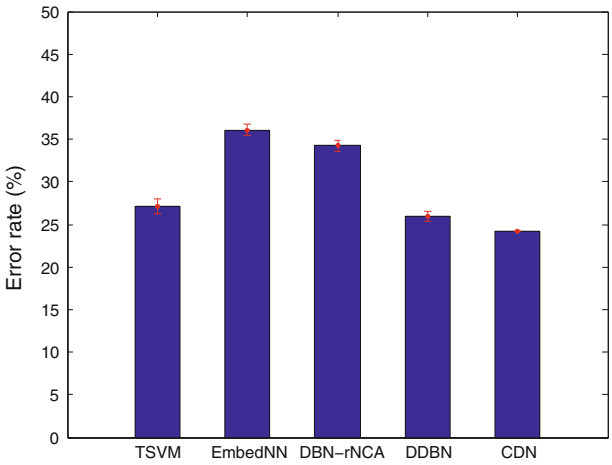


Fig. 5 Average and variance of error rates with 75 labeled data on Caltech 101

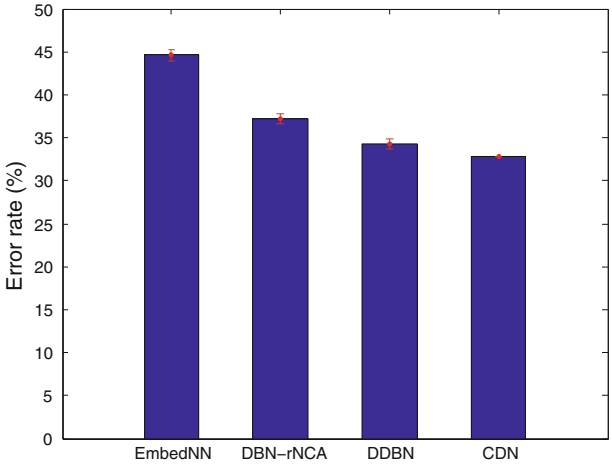


Fig. 6 Average and variance of error rates with 20 labeled data on MNIST

a large number of different initializations, the results of CDN with same number of labeled are same for 20 times running. Because of greedy layer-wise initialization, especially the abstraction of CRBM.

3.3 MNIST Dataset

MNIST is a large handwritten digit database containing 70,000 images with 10 classes. In this experiment, we use 100 labeled images and 70,000 unlabeled images for classification. The CDN structure used in this experiment is 6-24-500-10, which represents the number of groups in two convolutional hidden layers are 6 and 24, the number of units in one normal hidden layer and output layer are 500 and 10 respectively.

We set the number of labeled images equal to 20, 50, 100 respectively, the rest images are used as unlabeled data. Figures 6, 7 and 8 show the classification error rate of different

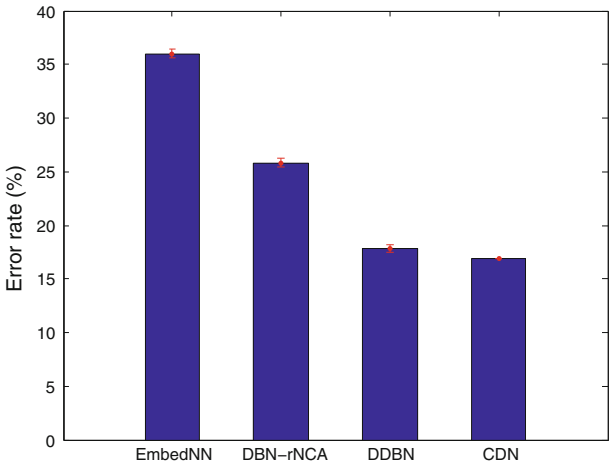


Fig. 7 Average and variance of error rates with 50 labeled data on MNIST

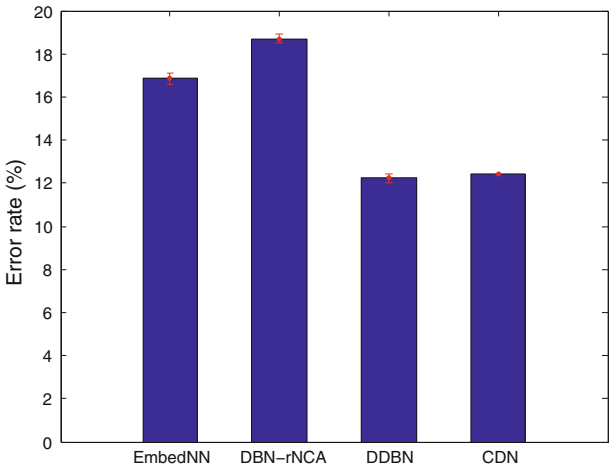


Fig. 8 Average and variance of error rates with 100 labeled data on MNIST

classifiers under different number of labeled images. For TSVM algorithm, the error rate is 16.81% when 100 labeled data and 2,000 unlabeled data are used [2]. However, due to the high computation cost, the experiment on TSVM has not finished for several weeks running when nearly 60,000 images are used as unlabeled data [13]. Through the table, we can see that CDN demos competitive performance comparing with other representative semi-supervised learning and deep learning methods.

4 Conclusions

In this paper, we propose a novel semi-supervised learning method, CDN, to address the image classification problem with few labeled images and large amount of unlabeled images. CDN seamlessly incorporate greedy layer-wise unsupervised learning method into the CNN

architecture, and use CRBM to abstract the image information effectively. One promising property of CDN is that it can effectively use the distribution of large amount of unlabeled data, together with few label information in a unified framework. In particular, CDN can greatly reduce the dimension of images through subsampling and abstracting the information of images through the cooperation of CRBM and RBM. Then an exponential loss function is used to refine the constructed deep architecture with few label information. Experiments conducted on two image datasets demonstrate that CDN outperforms most state-of-the-art semi-supervised learning algorithms on classification tasks. In future, we will continue to optimize the classification performance of CDN, study the performance of CDN for supervised learning, and use it for video classification.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (No. 61272383, No. 61173075 and No. 60973076).

References

1. Chapelle O, Scholkopf B, Zien A (2006) Semi-supervised learning. MIT Press, Cambridge
2. Collobert R, Sinz F, Weston J, Bottou L (2006) Large scale transductive SVMs. J Mach Learn Res 7:1687–1712
3. Desjardins G, Bengio Y (2008) Empirical evaluation of convolutional rbms for vision. Tech. rep
4. Feng GY, Hu DW, Zhou ZT (2008) A direct locality preserving projections (DLPP) algorithm for image recognition. Neural Process Lett 27(3):247–255
5. Hinton GE (2002) Training products of experts by minimizing contrastive divergence. Neural Comput 14(8):1771–1800
6. Hinton GE (2010) Learning to represent visual input. Philos Trans R Soc B-Biol Sci 365(1537):177–184
7. Hinton GE, Salakhutdinov R (2006) Reducing the dimensionality of data with neural networks. Science 313:504–507
8. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554
9. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551
10. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
11. Lee H, Grosse R, Ranganath R, Ng A (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: International conference on machine learning. ACM, Montreal, Canada, pp 609–616
12. Li F, Fergus R, Perona P (2004) Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: CVPR workshop on generative-model based vision. Elsevier Science Inc., New York, pp 1–9
13. Liu Y, Zhou S, Chen Q (2011) Discriminative deep belief networks for visual data classification. Pattern Recognit 44(10–11):2287–2296
14. Machajdik J, Hanbury A (2010) Affective image classification using features inspired by psychology and art theory. In: International conference on multimedia. ACM, New York, pp 83–92
15. Salakhutdinov R, Hinton GE (2007) Learning a nonlinear embedding by preserving class neighbourhood structure. J Mach Learn Res 2:412–419
16. Weston J, Ratle F, Collobert R (2008) Deep learning via semi-supervised embedding. In: International conference on machine learning. ACM, Helsinki, pp 1168–1175
17. Zhu X (2007) Semi-supervised learning literature survey. Tech. rep.. University of Wisconsin Madison, Madison