Neural Process Lett (2019) 49:305–319
https://doi.org/10.1007/s11063-017-9751-y

A Novel Gaussian–Bernoulli Based Convolutional Deep Belief Networks for Image Feature Extraction

Ziqiang Li[1] · **Xun Cai**[1] · **Yun Liu**[1] · **Bo Zhu**[1]

Published online: 12 March 2018

Abstract Image feature extraction is an essential step in the procedure of image recognition. In this paper, for images features extracting and recognizing, a novel deep neural network called Gaussian–Bernoulli based Convolutional Deep Belief Network (GCDBN) is proposed. The architecture of the proposed GCDBN consists of several convolutional layers based on Gaussian–Bernoulli Restricted Boltzmann Machine. This architecture can take the advantages of Gaussian-Binary Restricted Boltzmann machine and Convolutional Neural Network. Each convolutional layer is followed by a stochastic pooling layer for down-sampling the feature maps. We evaluated our proposed model on several image benchmarks. The experimental results show that our model is more effective for most of images recognition tasks with comparably low computational cost than some of popular methods, which is suggested that our proposed deep network is a potentially applicable method for real-world image recognition.

Keywords Deep learning · Feature extraction · Gaussian–Bernoulli based convolutional restricted Boltzmann machines · Gaussian–Bernoulli based convolutional deep brief network · Image classification

1 Introduction

Image recognition has been wildly used in many applications [1–3]. During the procedure of image recognition system, image feature extraction is one of the most important steps, which directly affects the accuracy of final results. It is known that the traditional image features are colour, shape, texture and so on. However, these raw features are not enough for image

✉ Xun Cai
caixunzh@sdu.edu.cn

Ziqiang Li
ziqiang_li@hotmail.com

1 School of Computer Science and Technology, Shandong University, Jinan 250101, Shandong, China

recognition. Nowadays, more abstractive features, such as Histogram of Oriented Gradient (HOG) [4] and Scale-Invariant Feature Transform (SIFT) [5] have been successfully used for some specific image recognition domains. However, as these hand-crafted features are fixed, they are not adaptive to many complex situations. With the development of deep learning in recent year, more general and efficient features are learned by this multi-level unsupervised deep learning methodology, which shows more applicable in visual recognition tasks.

Among these deep learning methods, Deep Belief Network (DBN) [6] is one of the most classical deep learning models, which consists of multiple cascaded Restricted Boltzmann Machines (RBM) [7]. There are many successful variants of RBM have been proposed in order to further improve feature representation ability for different types of data. For example, Conditional Restricted Boltzmann Machine (CRBM) [8] was proposed for collaborative filtering, Gaussian–Bernoulli Restricted Boltzmann Machine (GRBM) [9] was asserted to be more suitable for real-valued continuous data, and Recurrent Temporal Boltzmann Machine (RTBM) [10] was developed for representing high-dimensional sequence data, such as speech data and video data.

As above mentioned, GRBM is more suitable for real-valued data. Therefore, it is more applicable to model non-binary image data than tradition RBM. However, since the full-connected architecture of GRBM is inherited from traditional RBM, it is easily involved over-fitting in feature learning and led to large computation costs.

In recent years, Convolutional Neural Network (CNN) [11] have shown the extraordinary ability of image feature representation in image recognition field [12]. The architecture of CNN consists of several multi-level convolutions and poolings. By this architecture, CNN can efficiently extract the spatial information from images and greatly decrease the amount of computations.

In this paper, we propose a novel generative model called Gaussian–Bernoulli based Convolutional RBM (GCRBM) by combining CNN with GRBM. Moreover, by stacking the proposed GCRBMs, a corresponding deep model is also proposed, which is called Gaussian–Bernoulli based Convolutional Deep Belief Network (GCDBN). Taking advantages of CNN and GRBM, this deep architecture of GCDBN can extract meaningful features through a full-size image by generative convolution filters, which can reduce quite a number of connecting weights and can learn the spatial informations from neighboring image patches more effectively.

The rest of this paper is organized as follows: In Sect. 2, we briefly review the GRBM. Our proposed the GCRBM and its deep version GCDBN will be shown in Sects. 3 and 4, respectively. The pseudo-code of training procedure and computational complexity of GCRBM are given in Sect. 5. Experimental results of GCRBM and GCDBN are shown in Sect. 6 and some conclusions will be given in Sect. 7.

2 Related Work

GRBM is a generalized form of RBM of which the energy function includes two variant vectors **v** and **h** denoted corresponding visible and hidden units. If we use a weight matrix denoted by W to represent symmetric connections between two layers, the energy function of GRBM [9] is defined as

$$E(\mathbf{v}, \mathbf{h}) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i,j} W_{ij} h_j \frac{v_i}{\sigma_i} - \sum_j c_j h_j, \tag{1}$$

where real-valued states of *i*th visible unit and binary states of *j*th hidden unit are defined as v_i and h_j, respectively. b_i is the bias of *i*th visible unit, c_j is the bias of *j*th hidden unit, and σ denotes the standard deviation for Gaussian distribution. Then, the joint probability distribution of a state (**v**,**h**) was defined as

$$p\left(\mathbf{v},\mathbf{h}\right)=\frac{1}{Z}e^{-E(\mathbf{v},\mathbf{h})},\tag{2}$$

where Z is the partition function. Since GRBM is derived from RBM, the conditional probabilities of visible and hidden units are performed by Gibbs sampling and given as follows:

$$p\left(h_j=1|\mathbf{v}\right)=sigmoid\left(\sum_i W_{ij}\frac{v_i}{\sigma_i}+c_j\right)\tag{3}$$

and

$$p\left(v_i=v|\mathbf{h}\right)=\mathcal{N}\left(v\,;\sum_j h_jW_{ij}+b_i,\sigma_i^2\right),\tag{4}$$

where $\mathcal{N}\left(\cdot;\mu,\sigma^2\right)$ denotes the Gaussian distribution with mean μ and variance σ^2. v is reconstructed value with the Gaussian distribution.

3 Proposed GCRBM

3.1 Notation

The architecture of proposed GCRBM contained two main parts, as shown in Fig. 1. The first part is feature extraction where we create several feature maps in hidden layer which is denoted by H. Initially it is generated from visible layer which is denoted by V. The second part is image representation, where the created feature maps are zero-padded and convolved to reconstruct the visible layer.

Let the N_V, N_H and N_W denote the heights of each image, feature and convolutional filter, respectively. In feature extraction part, the visible layer has $N_V\times N_V$ units and a bias b. Each visible unit corresponds to each pixel of an image. Similarly, the hidden layer has $M\times N_H\times N_H$ units, corresponding to M *groups* of $N_H\times N_H$ feature maps. Each feature map corresponds to a $N_W\times N_W$ convolution filter and a bias c_k. Given the N_V and N_W, N_H = N_V-N_W+1. The pixels for each feature map in hidden layer are connected to the blocks of the image in visible layer though a generative filter.

In image reconstruction part, after zero-padding, the N_H of each feature map becomes N_V+N_W-1, which can ensure the size of reconstructed image equals to that of original image after convolution. Each reconstructed visible unit connected to m *groups* of feature maps through corresponding filters, respectively. For convenience, the size of the visible and hidden layer is fixed to be square. Convolution is denoted as $*$, dot production as $\bullet$, and the operation of flipping the array horizontally and vertically as $rot180°\left(\cdot\right)$.

3.2 Energy Function of GCRBM

Taking account of the convolutional filters between visible and hidden units, we modify the energy function of GRBM in Eq. (1) as follows:

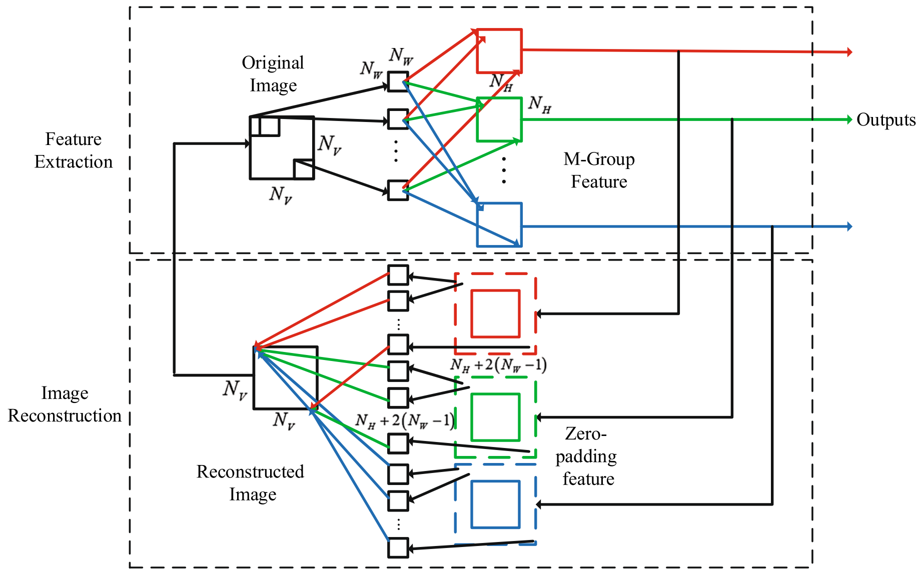

Fig. 1 The architecture of proposed GCRBM model

$$E(\mathbf{v},\mathbf{h}) = \sum_{i,j=1}^{N_V} \frac{(v_{ij}-b)^2}{2\sigma_{ij}^2} - \sum_{m=1}^{M}\sum_{i,j=1}^{N_H}\sum_{r,s=1}^{N_W} h_{ij}^m W_{rs}^m \frac{v_{i+r-1,j+s-1}}{2\sigma_{i+r-1,j+s-1}^2} - \sum_{m=1}^{M}\sum_{i,j=1}^{N_H} c_m h_{ij}^{km}. \tag{5}$$

Based on the definitions of operations described in Sect. 3.1, Eq. (5) can be simplified as follows:

$$E(\mathbf{v},\mathbf{h}) = \sum_{i,j} \frac{(v_{ij}-b)^2}{2\sigma_{ij}^2} - \sum_m h^m \bullet \left(rot180^\circ\left(W^m\right) * \frac{v}{2\sigma^2}\right) - \sum_m \sum_{i,j} c_m h_{i,j}^m. \tag{6}$$

3.3 Calculation of Conditional Probability

During feature extraction part, based on Eq. (2), the conditional probability of *ij*-th unit of *m*-th feature map is derived from Eq. (6) and is formulated as:

$$P\left(h_{ij}^m = 1\,|\mathbf{v}\right) = sigmoid\left(c_m + \left(rot180^\circ\left(W^m\right) * \frac{v}{2\sigma^2}\right)_{ij}\right). \tag{7}$$

Through Eq. (7), we can sample the value of hidden unit based on computed probability. Unlike [13], during image reconstruction step, we add zero-padding to ensure that each reconstructed image has the same size as that of original image. Thus, the Gaussian conditional probability distribution of *ij*-th visible unit is given as

$$P\left(v_{ij} = v|\mathbf{h}\right) = \mathcal{N}\left(v; b + \sum_m W^m * h^m, \sigma_i^2\right). \tag{8}$$

Based on Eq. (8), we obtain each pixel of reconstructed image. To prevent the Gibbs sampling from being infinite, we adopt the k-step Contrastive Divergence (CD) learning method as in [14] to estimate the gradient of parameters. The CD learning method effectively reduce the deviation of estimation when $k = 1$ [14].

3.4 Weight Updating

To speed up rate of updating m-th weight matrix, W^m, we introduce a new the momentum item W_i^m for the i-th iteration. The updating rule of W_i^m is formulated as follows:

$$W_i^m = \lambda W_{i-1}^m - \nabla W^m/(N_H + N_V) + \left\| W^m \right\|_2 / N_W^2 - W^m/(N_H + N_V), \tag{9}$$

where λ is the momentum factor for controlling the updating rate. ∇W^m represents the gradient of W^m. $\|\cdot\|_2$ denotes the L_2-norm for sparse penalty. Unlike the method proposed in [13] that reconstructs a part of visible units and uses a large sparse item, we normalize sparse penalty by N_W^2. The fourth term in Eq. (9) is used to prevent the training error from decreasing too fast which is called weight decay [12]. Then, W^m is updated as

$$W^m = W^m + W_i^m. \tag{10}$$

For calculating the ∇W^m, since the traditional C-RBM [13] is regardless of variance of training samples, we modify the gradient computation rules in [6] as:

$$\nabla W^m = \eta \left\langle \frac{rot180^\circ\,(h^m) * v}{2\sigma_i^2} \right\rangle_{data} - \eta \left\langle \frac{rot180^\circ\,(h^m) * v}{2\sigma_i^2} \right\rangle_{recon} \tag{11}$$

$$\nabla b = \eta \sum_i \left(\left\langle \frac{v_i}{2\sigma_i^2} \right\rangle_{data} - \left\langle \frac{v_i}{2\sigma_i^2} \right\rangle_{recon} \right) / N_v^2 \tag{12}$$

$$\nabla c_m = \eta \sum_j \left(\left\langle h_j^m \right\rangle_{data} - \left\langle h_j^m \right\rangle_{recon} \right) / N_h^2 \tag{13}$$

$$\begin{aligned} \nabla \sigma_{ij} = {} & \eta \left\langle \frac{(v_{ij} - b)^2}{\sigma_{ij}^3} - \sum_{m=1}^{M} h^m \bullet rot180^\circ(W^m) * \frac{v}{2\sigma^2} \right\rangle_{data} \\ & - \eta \left\langle \frac{(v_{ij} - b)^2}{\sigma_{ij}^3} - \sum_{m=1}^{M} h^m \bullet rot180^\circ(W^m) * \frac{v}{2\sigma^2} \right\rangle_{recon} \end{aligned} \tag{14}$$

where η represents learning rate, $\langle\cdot\rangle_{data}$ and $\langle\cdot\rangle_{recon}$ denote the desired values of data distribution and reconstruction distribution, respectively. ∇ denotes the gradient of each following parameter.

4 Stacking GCRBMs to Construct GCDBN

It is known that the deep model has more effective ability of feature extraction than that of the shallow one. Hence, by stacking the GCRBM, we proposed a novel deep model called Gaussian–Bernoulli based Convolutional Deep Belief Network (GCDBN). The general architecture of GCDBN is shown in Fig. 2. In this model, several GCRBMs are connected to enhance the extraction ability of GCDBN. All of features extracted are mapped into a dense vector to obtain distributed representation in the dense layer before the final Softmax

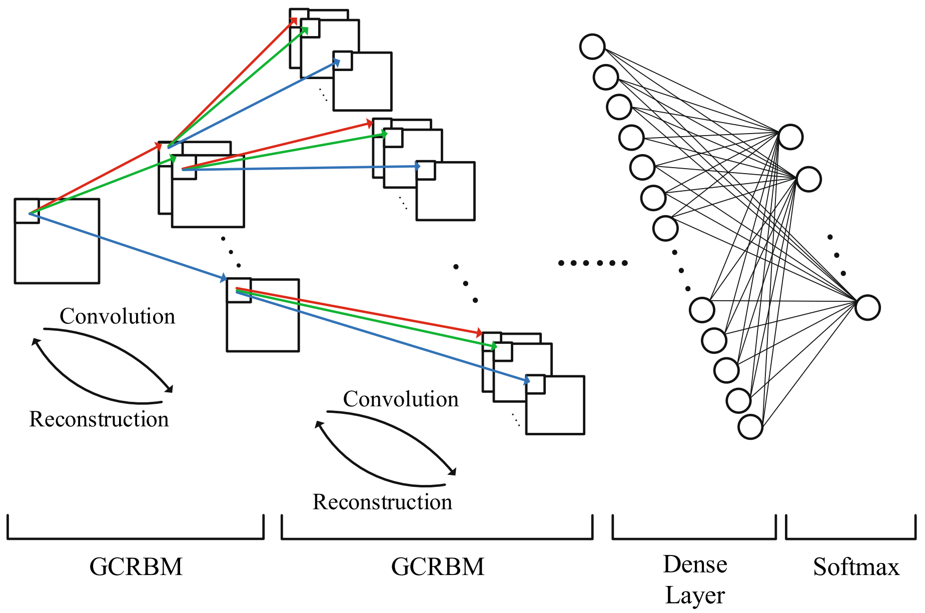

Fig. 2 The architecture of proposed GCDBN model

classification. To fine-tune this model, instead of using standard back-propagation which will easily lead to a slow convergence learning speed, we introduce the second order back-propagation Stochastic Diagonal Levenberg Marquardt (SDLM) algorithm [15] by computing the learning rate of current iteration to accelerate the speed of convergence.

5 Analysis

5.1 Training Procedure for Proposed GCRBM

The pseudo-code of training procedure of GCRBM by CD-k strategy is given in Alg. 1. The aim of Alg. 1 is utilizing input images and initially setting parameters like learning rate η, momentum item λ, and the number of filters m to obtain the filter set W, filter bias c, input bias b, and well-trained variance σ^2. We divided the total training procedure into three phase: the positive phase, the negative phase and the updating phase. In the positive phase, each image is convoluted by using Eq. (7) to generate m feature maps. Similarly, in the negative phase: the m feature maps are convoluted by using Eq. (8) to reconstruct a generated image. The gradients of each parameter and filters are calculated by Eqs. (11), (12), (13), (14). Finally, in the updating phase, the W is updated by proposed updating rules Eqs. (9), (10), meanwhile filter bias b, image bias c and variance σ^2 are updated by gradient descent.

However, it is not means the only CD-k strategy can be applied in GCRBM. In fact, other RBM based training methods such as Persistent CD-k (PCD-k) [16] and Parallel Tempering (PT) [17] are also can be used in GCRBM. For PCD-k training strategy, differing with CD-k, the initial values of Gibbs chain are given by the pixels of feature maps. Hence, The k-th Gibbs-sampling of PCD in GCRBM starts from H_{gaven} instead of the V_{data}. For Parallel Tempering strategy, it requires to set L Gibbs chains according to different temperatures

Algorithm 1 training procedure of GCRBM by CD-k strategy

```
Input: an original image $V_{data}$, variance $\sigma^2$, learning rate $\eta$, momentum item $\lambda$, the number of filters $m$, the
  iteration $k$.
Output: filter set $W$, filter bias $c$, input bias $b$, well-trained variance $\sigma^2$.
  Initialization: $W, c, b, \sigma^2$.
  for $s = 1 \to k$ do
    Positive Phase:
    for $i = 1 \to m$ do
      Calculate $P(H^i_{data})^s$ using Eq. (7)
      $H^s_{data} = Bernoulli\ (P(H^i_{data})^s)$
    end for
    Negative Phase:
    Calculate $P(V_{recon})^s$ using Eq. (8)
    $V^s_{recon} = Gaussian\ (P(V_{recon})^s)$
    for $i = 1 \to m$ do
      Calculate $P(H^i_{recon})^s$ using Eq. (7)
      $H^s_{recon} = Bernoulli\ (P(H^i_{recon})^s)$
    end for
  end for
  Updating Phase:
  Calculate gradient of $W, b, c, \sigma^2$ using Eqs. (11), (12), (13), (14)
  Update $b, c, \sigma^2$ with learning rate $\eta$
  Update $W$ with learning rate $\eta$ and momentum $\lambda$ using Eqs. (9), (10).
```

defined as $0 \leq T_0 < T_1 < \cdots < T_L \leq 1$. Let keep positive phase of PT as same as that of CD-k. In the negative phase of PT, l-th Gibbs chain operates Gibbs sampling to obtain $V^s_{recon^{T_l}}$ and $H^s_{recon^{T_l}}$. After sampling, every pair of sampled values will be swapped according to swapping probability. The swapping probability $P_{swapping}$ is given as:

$$P_{swapping} = min\left\{1, \left(\frac{1}{T_l} - \frac{1}{T_{l-1}}\right) exp\left(E_{temp}\right)\right\}, \tag{15}$$

where the E_{temp} is defined as the energy difference between $E\left(V^s_{recon^{T_l}}, H^s_{recon^{T_l}}\right)$ and $E\left(V^s_{recon^{T_{l-1}}}, H^s_{recon^{T_{l-1}}}\right)$, which is formulated as:

$$E_{temp} = E\left(V^s_{recon^{T_l}}, H^s_{recon^{T_l}}\right) - E\left(V^s_{recon^{T_{l-1}}}, H^s_{recon^{T_{l-1}}}\right). \tag{16}$$

Finally, the $V^s_{recon^{T_L}}$ and $H^s_{recon^{T_L}}$ under the $T_L = 1$ are the outputs of negative phase. The updating phase of PT is same with CD-k.

5.2 Computational Complexity

For a three layers GCDBN which has a GCRBM, a dense layer and the softmax. The computational complexity of constructing the GCDBN model can be calculated mainly into two stages. The first stage is the training GCRBM, and the second stage is fine-tuning the GCDBN. Here, we choose the CD-k strategy to train GCRBM. The biggest computational cost of training GCRBM is convolution. Since the convolution operates in feature extraction part and image recognition part, the total computational complexity of convolution is $O\left(km\,(N_W)^2\left((N_H)^2 + (N_V)^2\right)\right)$.

For fine-tuning GCDBN, we utilized the SDLM which is described in Sect. 4 to train dense layer and softmax. Supposing the number of neurons and classification in dense layer and softmax is N_d and C, respectively. The epochs of fine-tuning GCDBN is M. The total computational complexity of calculating the Hessian matrix of SDLM is $O\left(M\left(N_d\right)\left(m\left(N_H\right)^2+C\right)\right)$. Hence, the overall computational complexity of GCDBN with N training images is

$$O\left(Nkm\left(N_W\right)^2\left(\left(N_H\right)^2+\left(N_V\right)^2\right)+MN\left(N_d\right)\left(m\left(N_H\right)^2+C\right)\right). \tag{17}$$

Let GDBN keep the same number of neurons and training epochs with GCDBN. The computational complexity of GDBN is $O\left(MN\left(m(N_H N_V)^2\right)\right)$. We can find that the computational complexity of GCDBN is lower than that of GDBN. Especially, when using large input images as input data ($N_V >> N_W$), the advantage of GCDBN will be more obvious. The reason is that the GCDBN uses convolution to replace the fully-connected structure, which can greatly reduce the computational consumption in feature extraction.

For CNN (LeNet-2), the overall computational complexity without pre-training is $O\left(MN\left(m\left(N_W N_H\right)^2\left(N_d\right)\left(m(N_H)^2+C\right)\right)\right)$. The gradient descent fixed the learning rate for every epoch, but the SDLM calculated the fittest learning rate for every epoch. Hence the proposed GCDBN can take less training epochs and times than CNN, which ensures the proposed GCDBN is time efficient.

5.3 Differences Between GCDBN and Other RBM-based Models

There are several differences between our proposed model and several state-of-the-art RBM-based models:

First, comparing with GRBM [9], our proposed model used convolutional filters to replace fully-connected weights. In this way, computational costs of our proposed model is much lower than that of GRBM and neighborhood space relations of input images are preserved well.

Second, considering the variances of real-valued images in training procedure, our proposed GCDBN introduced Gaussian–Bernoulli process and updated variances continuously during every epoch while C-RBM [13] and CDBN [18]do not. Since each pixel value in real-valued input images are not binary. Gaussian–Bernoulli process translated input pixel values to Gaussian distribution which could help feature maps represent original images distinguishably (see details in Sect. 6.5.2).

Algorithmic differences are also appeared in the training procedure of the proposed GCDBN. Most previous RBM-based works just adopted CD-1 training method, however, our work compared energy values of GCRBM by using CD-1, CD-10, PCD-1, PCD-10 and PT-10 training methods and found the best way for training GCRBM is PCD-1 (see details in Sect. 6.3). To speed up the convergence of loss-function and avoid over-fitting, CDBN employed a sparse variant to regulate sparsity of feature maps, which needs updating sparse variant in every epoch. C-RBM employed a sparsifying logistic non-linearity function to control the sparsity of hidden units and accelerate the convergence with two parameters. However, in our work, under the PCD-1 training method, we used proposed updating rule to train GCRBM in Sect. 3.4. The proposed updating rule combines L_2-norm and momentum, which is demonstrated effectiveness to GCRBM (see details in Sect. 6.5.1).

6 Experiments

To evaluate the performance of the proposed model, we implemented several experiments on two image benchmarking datasets and compared the results of proposed GCRBM with those of RBM [7], GRBM [9], Auto-encoder (AE) [19], C-RBM [13] and CNN [11]. In these models, features were extracted by all the models and fed into a linear SVM classifier for recognition. Additionally, we also compared accuracy rate of our proposed GCDBN with that of some state-of-the-art deep models to demonstrate the effectiveness of our proposed deep model.

6.1 Datasets Description

In this paper, we selected two benchmarking datasets for our experiments which are Fifteen Scene Category dataset (Scene-15) and CIFAR-10 dataset, respectively. Scene-15 dataset contains 4485 different scene images which are categorized into 15 classes. CIFAR-10 dataset is one of the most challenging datasets of object recognition, which contains 60,000 colorful natural images equally separated into 10 classes. Both datasets involve complex background, different levels of illumination and occlusion.

6.2 Model Preprocessing

For Scene-15 dataset, we manually resized the total samples to be 128 $\times$128, while we remained the original size for CIFAR-10 dataset. All the images in each dataset were whitened by Zero-phase Component Analysis (ZCA) [14] in order to eliminate the correlation of neighbouring pixels. We constructed a 5-layer GCDBN in which the first and the third layer are convolution layers, the second and the fourth layer are pooling layers and the fifth layer is dense layer. In the convolution layers, the proposed GCRBMs are used for training convolutional filters. In the pooling layer the probabilistic max pools [18] with equally sized pooling units of 2×2 are set to down sampling feature maps. In dense layer, feature are distributedly represented before sending to the Softmax classifier.

For GCRBM, GCDBN and GRBM, as all the samples are whitened, the initial σ is set to be 1. For RBM and AE, the number of hidden units is set to be 1000. For CNN, we applied the classical architecture of LeNet-2 and LeNet-5. For LeNet-2 and C-RBM, there are eight 9×9 filters in the convolution layer and the pooling region is of size 2×2 in the pooling layer. For LeNet-5, the first and the third layers are convolution layers, the second and the fourth layers are pooling layers. The numbers of filters in the first layer and the third layers are 8 and 24, respectively. The size of each filter is fixed to be 9×9. According to [14], weights are randomly initialized by $U \sim \left(-\frac{\sqrt{6}}{\sqrt{N_H}+\sqrt{N_V}}, \frac{\sqrt{6}}{\sqrt{N_H}+\sqrt{N_V}}\right)$ and the biases are initialized by zero. We fixed learning rate η and momentum factor λ to be 0.05 and 0.9 respectively.

6.3 Selection of Training Methods

In order to find the best learning method for the proposed GCRBM, we evaluated five popular learning methods on Scene-15 dataset which are called 10-step Parallel Tempering (PT-10) [17], 1-step and 10-step Persistent Contrastive Divergence (PCD-1, PCD-10) [16], 1-step and 10-step Contrastive Divergence (CD-1, CD-10), respectively. The experimental results are listed in Fig. 3. It shows that the PCD-1 performed best among the five and CD-10 performed worst. The reasons might be that when using the CD-k and the PT-k algorithm, every training epoch begins at the input data. But when using the PCD-k, the next training epoch begins at the hidden units of last training epoch. The obtained hidden units as input data have the better capability of convergence for GCRBM proposed model. While uses k Gibbs chains to enhance the capability of convergence. In addition, computational cost of PCD-1 is just 0.25 times of PT-10. In summary, the PCD-1 is the best choice to train the GCRBM for image classification.

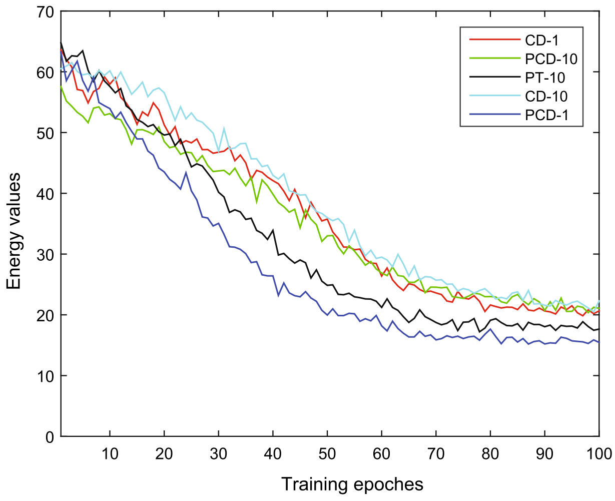

Fig. 3 Energy values of GCRBM with different training methods on Scene-15 dataset. We test the PCD, CD with k = 1, 10 and PT-10. For PT-10, temperature is adjusted from 0.1 to 1 with 0.1 interval. The learning rate and momentum factor are set to be 0.1 and 0.9, respectively

Table 1 The reconstructed errors of filters with different size from 5 × 5 to 12 × 12

Filter size	$N_W = 5$	$N_W = 6$	$N_W = 7$	$N_W = 8$	$N_W = 9$	$N_W = 10$	$N_W = 11$	$N_W = 12$
R.e	1.283	1.257	1.249	1.216	1.179	1.201	1.212	1.243

6.4 Selection of Best Suitable Size of the Filters

In order to find the most suitable size of the filters, we evaluated reconstructed errors of GCRBMs with the different size of filters from 5 × 5 to 12 × 12 on Scene-15 dataset. Based on Sect. 6.3, we adopted PCD-1 as training method. The reconstructed errors are listed in Table 1. It is obvious that when $N_W = 9$, the reconstructed error is minimal. Additionally, filters with too large or too small size will lead to large the reconstructed errors due to feature loss.

6.5 Evaluation of Performance

6.5.1 Cross Validation

Under the PCD-1 training method, we compared the five folds cross-validation errors of GCRBM using proposed weight updating rule with that of using original rule [6]. The total images of Scene-15 dataset divided into training set with 4000 images and validation set with least 485 images. The error rates with epochs are shown in Fig. 4.

We can find that even though the error rate of original updating rule drops lower than that of proposed updating rule on training set, for validation, the error rate of proposed method is lower than original one. The theoretical reason of this phenomenon is that we added L_2 norm in the proposed updating rule. In order to penalize some big weights which misleadingly represents some learned features, the added L_2-norm can ensure the each weight of filter W^m

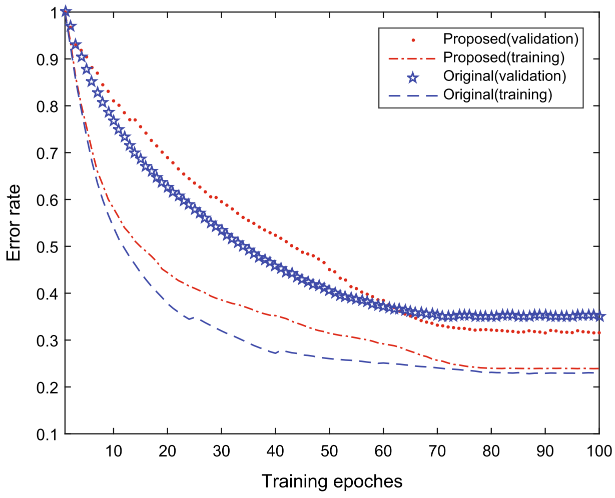

Fig. 4 The convergence of error rates of proposed weight updating rule and original weight updating rule under PCD-1 learning method with training epochs

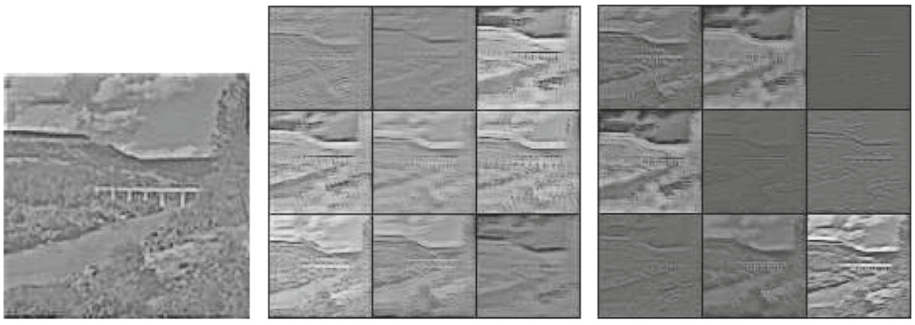

Fig. 5 **a** Image shows the whitened original random image. **b** Image shows the 9 pooled features by proposed GCRBM. **c** Image shows the 9 pooled features by CRBM [18]

has Relatively small value, which has been demonstrated the effectiveness of reducing over-fitting phenomenon in some literatures [20]. Hence, the outputs could be more generalized rather than leaded by certain learned features. Therefore, it is concluded that the proposed updating rule effectively can avoid over-fitting and obtain less test error.

6.5.2 Capability of Feature Representation

Based on PCD-1 training methods, for scene-15 dataset, we compare 9 pooled features by proposed GCRBM and CRBM in Fig. 5b and c, which were generated by 9 convolutional filters with size of 9×9 and pooled by 2×2 probabilistic max pool. As shown in Fig. 5, the features of GCRBM are more distinguishable and unique than those of CRBM, which not only exhibits diversities of generated features according to certain input images, but also obtains better recognition accuracy than CRBM (see details in Sect. 6.5.3).

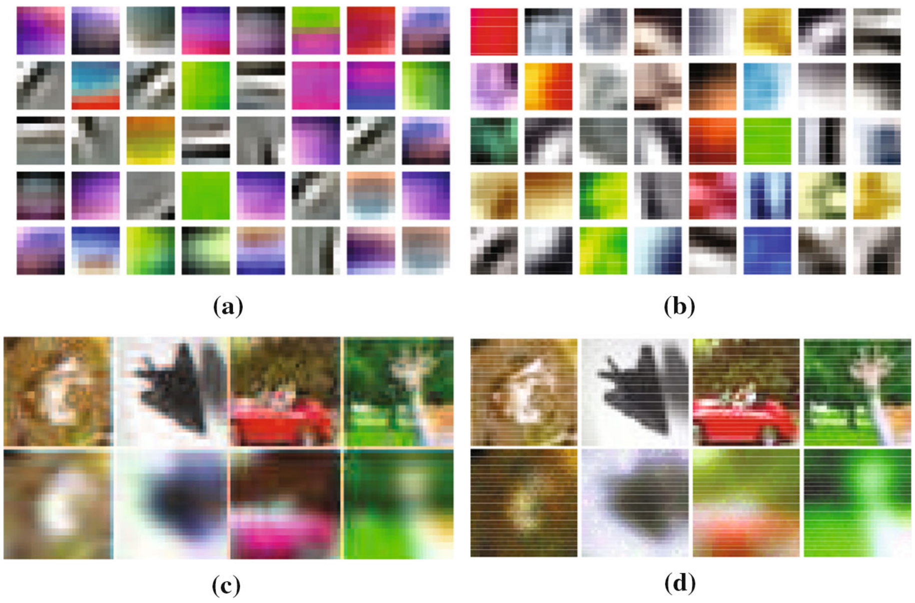

Fig. 6 **a** 40 learned filters in GCDBN after 500 training epochs. **b** 40 learned filters in CDBN after 500 training epochs. **c** *top row*: 4 ramdom original images; *bottom row*: 4 reconstructed images in GCDBN corresponding to original images. **d** *top row*: 4 ramdom original images; *bottom row*: 4 reconstructed images in CDBN corresponding to original images

For CIFAR-10 dataset, in order to show differences on feature representation between GCDBN and CDBN [18], 40 well-learned filters and 4 randomly selected reconstructed images of GCDBN and CDBN with their corresponding original images are shown in Fig. 6. It shows that Filters of GCDBN and CDBN learned different details from original images. However, colours and edges of each filter in GCDBN are learned more clearly than that of CDBN. Besides, It can be easily seen that similarity between reconstructed images and original images of GCDBN is higher than that of CDBN. The reasons could be explained that Guassian–Bernoulli process benefits the reconstruction in GCDBN. More specifically, the variance changing in training process of GCDBN helps filters to learn colorful and edged details better than original training process of CDBN. That is also the most significant difference between proposed GCDBN and CDBN.

6.5.3 Recognition Accuracy

For Scene-15 dataset, we set the numbers of filters in the first and the third layer of the proposed GCDBN to be 9 and 24 respectively and the number of neurons in the dense layer is fixed to be 2000. The recognition accuracy of our proposed model compared with that of RBM, GRBM, AE, CNN and some state-of-the-art models is shown in Table 2. It is obvious that accuracy rate of our proposed GCRBM is better than RBM and its derived models. Meanwhile, the accuracy rate of our proposed deep model, GCDBN, is better than those of SPM [21], CA-TM [22]. Only C-Bow [23] and CNN (fusion) [24] outperformed our proposed GCDBN. However, the C-Bow and CNN (fusion) pre-trained on PASCAL 2007 dataset and ImageNet 2012 dataset, respectively. While the GCDBN did not have any pre-train procedures. For time efficiency, the CNN (fusion) reported that it integrated 4 AlexNets

Table 2 Recognition accuracy on Scene-15 dataset

Methods	Accuracy (%)	Methods	Accuracy (%)
RBM [7]	58.79	SPM [21]	81.10
GRBM [9]	65.12	CA-TM [22]	82.50
AE [19]	67.87	C-BoW [23]	87.80
C-RBM [13]	71.14	CNN (fusion) [24]	90.70
CNN (LeNet-2) [11]	73.40	**GCRBM + Softmax**	75.33
CNN (LeNet-5) [11]	79.17	**GCDBN + Softmax**	85.96

Table 3 Recognition accuracy on CIFAR-10 dataset

Methods	Accuracy (%)	Methods	Accuracy (%)
RBM [7]	56.12	SPN [25]	83.96
GRBM [9]	63.68	Stochastic pooling [26]	84.87
AE [19]	60.45	MCDNN [27]	88.79
C-RBM [13]	67.43	NIN [28]	91.20
CNN (LeNet-2) [11]	72.97	**GCRBM + Softmax**	76.81
CNN (LeNet-5) [11]	86.65	**GCDBN + Softmax**	88.41

and cost 7 days for training, but our proposed CGDBN just cost around 4 h which included 2 CGRBM training and fine-tuning. We believe that the accuracy rate of the GCDBN will be further improved if more training data are provided and more convolutional layers are added.

For CIFAR-10 dataset, we fixed the number of filters in the first and the third layer of GCDBN to be 16 and 40, respectively. And the number of neuron in dense layer is set to be 2000. Besides, we set 40 filters in GCRBM. The classification results are shown in Table 3. The accuracy of the proposed GCRBM increases by 9.38% of GRBM and by 3.84% of LeNet-2, respectively. Moreover, our proposed deep model, GCDBN, outperforms some state-of-the-art methods such as Sum-Product Network (SPN) [25] and Stochastic Pooling [26]. The accuracy rates of Multi-Column DNN (MCDNN) [27] and Network in Network (NIN) [28] are better than ours. However, the MCDNN and NIN own more complex structures than ours. Especially, the NIN employed Data Augmentation to extend the numbers of training samples. Hence, we believe that the accuracy of the proposed GCDBN will be better if we use some training tricks or cascade more GCDBNs into one model in the future.

7 Conclusion and Future Work

In this paper, first, we present a novel probability generation model, GCRBM, based on Gaussian–Bernoulli Restricted Boltzmann Machine. The proposed model combines CNN model with GRBM by taking the advantages of the effectiveness of GRBM for learning discriminative image features and the capability of CNN for learning spatial features. After features extraction and probabilistic max pooling, a group of shift-invariance features are fed into a discriminative classifier.

Second, we propose a novel deep model GCDBN by stacking the proposed GCRBM. Several convolution and pooling layers are connected in the architecture of GCDBN, which can not only generate low-dimensioned and translation-invariant features, but also ensure effective ability of feature extraction. In addition, we use proposed updating rule and SDLM algorithm to replace traditional BP in fine-tuning GCDBN, which can not only speed up the convergence of entire model, but also reduce the over-fitting phenomenon.

From the experimental results on two benchmarking datasets, we can conclude that GCRBM can effectively improve the performance of feature extraction by introducing variance and convolution. In addition, it is also demonstrated the effectiveness of GCDBN in image classification. However, limited by computing environment, we just constructed one GCDBN which only has 5 layers. Hence, if we can add more convolutional layer and pooling layer in the GCDBN, the recognition accuracy will be improved.

In the future, we will further improve the ability of feature extraction of GCRBM and research new methods for improving training GCDBN. In addition, with cascading more GCDBN into one model, we expect the better accuracy rate than present way.

Acknowledgements This work was supported by National Natural Science Foundation of China under Grant No. 51473088 and National Key Research and Development Plan of China under Grant No. 2016YFC0301400.

References

1. Yu J, Tao D, Wang M, Rui Y (2015) Learning to rank using user clicks and visual features for image retrieval. IEEE Trans Cybern 45(4):767–779
2. Yang X, Liu W, Tao D, Cheng J (2017) Canonical correlation analysis networks for two-view image recognition. Inf Sci 385–386:338–352
3. Liu W, Zha Z, Wang Y, Lu K, Tao D (2016) p-Laplacian regularized sparse coding for human activity recognition. IEEE Trans Ind Electron 63(8):5120–5129
4. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE international computer vision and pattern recognition (CVPR). IEEE, pp 886–893
5. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60:91–110
6. Hinton G, Salakhutdinov R (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
7. Rumelhart DE, McClelland JL, PDP Research Group (1986) Parallel distributed processing: explorations in the microstructure of cognition, Vol 1–2
8. Salakhutdinov R, Mnih A, Hinton G (2007) Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the 24th international conference on machine learning. ACM, pp 791–798
9. Wang N, Melchior J, Wiskott L (2014) Gaussian-binary restricted Boltzmann machines on modeling natural image statistics. arXiv preprint arXiv:1401.5900
10. Sutskever I, Hinton GE, Taylor GW (2008) The recurrent temporal restricted Boltzmann machine. In: Advances in neural information processing systems, pp 1601–1608
11. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
12. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
13. Norouzi M, Ranjbar M, Mori G (2009) Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning. In: CVPR
14. Hinton G (2010) A practical guide to training restricted Boltzmann machines. Momentum 9(1):926
15. LeCun Y, Bottou L, Orr GB, Müller K-R (2012) Efficient backprop. In: Neural networks: tricks of the trade. Springer, pp 9–48

16. Tieleman T (2008) Training restricted Boltzmann machines using approximations to the likelihood gradient. In: Proceedings of the 25th international conference on machine learning, ICML 2008. ACM Press, New York, pp 1064–1071
17. Cho KH, Raiko T, Ilin A (2013) Enhanced gradient for training restricted boltzmann machines. Neural Comput 25(3):805–831
18. Lee H, Grosse R, Ranganath R, Ng AY (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: ICML
19. Bengio Y, Lamblin P, Popovici D, Larochelle H, Montral UD, Qubec M (2007) Greedy layer-wise training of deep networks. In: NIPS. MIT Press
20. Li SZ, Jain A (2009) Encyclopedia of biometrics. Springer, Berlin, p 883
21. Lazebnik S, Schmid C, Ponce J (June 2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: IEEE conference on computer vision and pattern recognition, vol 2. New York, pp 2169–2178
22. Niu Z, Hua G, Gao X, Tian Q (2012) Context aware topic model for scene recognition. In: IEEE conference on computer vision and pattern recognition. Providence, RI, pp 2743–2750
23. Su Y, Jurie F (2011) Visualword disambiguation by semantic contexts. In: ICCV
24. Koskela M, Laaksonen J (2014) Convolutional network features for scene recognition. In: ACM Multimedia
25. Gens R, Domingos P (2012) Discriminative learning of sum-product networks. Adv Neural Inf Process Syst 25:3248–3256
26. Zeiler MD, Fergus R (2013) Stochastic pooling for regularization of deep convolutional neural networks. Technical Report, New York University, arXiv:1301.3557
27. Ciresan DC, Meier J, Schmidhuber J (2012) Multi-column deep neural networks for image classification. In: CVPR
28. Lin M, Chen Q, Yan S (2014) Network in network. In: ICLR