Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Gaussian Cardinality Restricted Boltzmann Machines

Cheng Wan[†], **Xiaoming Jin**[†], **Guiguang Ding**[†] and **Dou Shen**[‡]
†School of Software, Tsinghua University, Beijing, China
‡Baidu Corporation, Beijing, China
wanc12@mails.tsinghua.edu.cn, {xmjin, dinggg}@tsinghua.edu.cn, doushen@gmail.com

Abstract

Restricted Boltzmann Machine (RBM) has been applied to a wide variety of tasks due to its advantage in feature extraction. Implementing sparsity constraint in the activated hidden units is an important improvement on RBM. The sparsity constraints in the existing methods are usually specified by users and are independent of the input data. However, the input data could be heterogeneous in content and thus naturally demand elastic and adaptive settings of the sparsity constraints.

To solve this problem, we proposed a generalized model with adaptive sparsity constraint, named Gaussian Cardinality Restricted Boltzmann Machines (GC-RBM). In this model, the thresholds of hidden unit activations are decided by the input data and a given Gaussian distribution in the pre-training phase. We provide a principled method to train the GC-RBM with Gaussian prior. Experimental results on two real world data sets justify the effectiveness of the proposed method and its superiority over CaRBM in terms of classification accuracy.

Introduction

Restricted Boltzmann Machine (RBM) has been widely studied since 2006 (Hinton and Salakhutdinov 2006; Hinton, Osindero, and Teh 2006; Hinton 2010). It has been applied to a variety of tasks such as image classification and voice recognition. As a building block of deep structure (Bengio et al. 2007; Salakhutdinov and Hinton 2009; Cho, Raiko, and Ilin 2011), it provides good initialization on both supervised and unsupervised learning (Lee et al. 2009; Snoek, Adams, and Larochelle 2012). One important research direction is to improve its performance, along which enforcing sparsity constraints has been an effective way.

There are several approaches to introduce sparsity to RBM. Implementing sparsity constraints as penalty functions (Lee, Ekanadham, and Ng 2007; Luo et al. 2011) is one of those approaches. The work by Goh et al. (Goh et al. 2010; 2011) is also based on the same idea and aims to obtain a more precise control of the regularization. Another important way is to integrate strict sparsity constraints directly into energy function, which is not recommended before the Cardinality-RBM (CaRBM) (Swersky et al. 2012) since it often results in non-trivial dependencies between hidden units that make inference intractable.

Swersky et al. presented CaRBM in (Swersky et al. 2012). The main idea of CaRBM is that no more than a given number of hidden units are activated simultaneously. To do so, a universal threshold of the number of activated hidden units is added to the joint probability distribution. Therefore, all hidden units compete with each other for the limited chances of being activated, which lead to the consequence that only the most important hidden units, in terms of representing input data, can be activated. This would make the model capable to obtain genuinely sparse representations.

However, it is improper to assume that all input data have the same threshold of the number of activated hidden units, in many real world applications. Naturally, data (for example, images, in our experiments) in different catalogs should activate different numbers of hidden units. Consider that if we use a high threshold to model data which actually activate a few hidden units, it would bring some redundancy. And if we use a low threshold to model data which actually activate plenty of hidden units, it would lead to some information loss. Therefore, it is necessary that adapt the thresholds to the input data.

In this paper, we propose a principled method, which replaces the universal threshold in CaRBM by thresholds sampled from a certain distribution. Our model is capable of handling the input data more flexibly. For example, in our experiments, the threshold of the number of activated hidden units should be higher when the input image is relatively more complex. We also analyze how parameters in our model influence the performance. Here we name the model Gaussian Cardinality RBM (GC-RBM) since we use Gaussian distribution in our experiments. The reasons for why we choose Gaussian distribution are elaborated. Experimental results show the advantage of GC-RBM in classification tasks compared with CaRBM and the improvement is extremely statistically significant.

Background

Restricted Boltzmann Machines

Restricted Boltzmann Machine is a particular type of energy-based model with hidden variables (Bengio 2009) . It could be represented by an undirected bipartite graph. The

joint probability of visible units **v** and hidden units **h** is given by:

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} exp(\mathbf{v}^\top W\mathbf{h} + \mathbf{v}\mathbf{b} + \mathbf{h}^\top \mathbf{c}), \quad (1)$$

where Z is the partition function, and the model parameters $W \in \mathbb{R}^{N_v \times N_h}$ represents visible-to-hidden interaction term, $\mathbf{b} \in \mathbb{R}^{N_v}, \mathbf{c} \in \mathbb{R}^{N_h}$ represents the visible and hidden biases respectively, and N_v, N_h represents the number of visible units and hidden units respectively.

The analytical solutions of the joint probability are difficult to be obtained due to intractability of the partition function. However, $P(h_i \mid \mathbf{v})$ are conditionally independent to each other (so as $P(v_i \mid \mathbf{h})$), which make that both $P(\mathbf{h} \mid \mathbf{v})$ and $P(\mathbf{v} \mid \mathbf{h})$ are tractable. Therefore the conditional distribution can be easily obtained from Eq.2:

$$P(h_i = 1 \mid \mathbf{v}) = \frac{e^{c_i + W_i \mathbf{v}}}{1 + e^{c_i + W_i \mathbf{v}}} = sigm(c_i + W_i \mathbf{v}), \quad (2)$$

With the conditional probability, Monte Carlo Markov Chain (MCMC [15]) sampling can be applied in order to obtain a stochastic estimator of the log-likelihood gradient of $P(\mathbf{v})$.

Gibbs sampling could give an unbiased stochastic estimator of the log-likelihood gradient. However, it takes too much time to converge. As an approximation method, Contrastive Divergence (CD) (Hinton 2002) could give an approximate result faster.

Cardinality Restricted Boltzmann Machines

CaRBM introduces the sparsity directly on the joint distribution of visible variables and hidden variables, which is given by the equation:

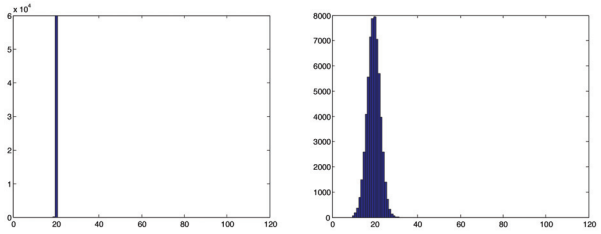
$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} exp(\mathbf{v}^\top W\mathbf{h} + \mathbf{v}\mathbf{b} + \mathbf{h}^\top \mathbf{c}) \cdot \psi_k \left(\sum_{j=1}^{N_h} h_j \right), \quad (3)$$

where $\psi_k(c)$ is assigned to value 1 only when $c \leqslant k$, and 0 others. And ψ_k is a *cardinality potential* (or *count potential*) (Gupta, Diwan, and Sarawagi 2007; Tarlow, Givoni, and Zemel 2010) which is related to the number of activated hidden units only. It also can be seen as making hidden units compete with each other since there are no more that k hidden units can be activated simultaneously.

As we know, hidden units are conditionally independent to each other in standard RBM when given the visible units. However, the hidden units in CaRBM have correlation with each other when cardinality potential is considered. So the conditional distribution $P(\mathbf{h} \mid \mathbf{v})$ is no longer factorized. But it would be easier if we convert it to be chain-structured:

$$P(\mathbf{h}, \mathbf{z} \mid \mathbf{v}) = \frac{1}{Z} \prod_{j=1}^{N_h} p(h_j \mid \mathbf{v}) \prod_{j=2}^{N_h} \gamma(h_j, z_j, z_{j-1}) \cdot \psi(z_N) \quad (4)$$

In this view, $\mathbf{z}$ is a N_h dimensional vector, where z_j can be seen as auxiliary variables deterministically related to the h variables by setting $z_j = \sum_{i=1}^{j} h_i$ and represents the cumulative sum of the first *j* hidden units. $\gamma(h_j, z_j, z_{j-1})$ is a deterministic "addition potential", which assigns the value one to any triplet $(\mathbf{h}, \mathbf{z}, \mathbf{z}')$ satisfying $\mathbf{z} = \mathbf{h} + \mathbf{z}'$. According to Eq.4, with the chain-structured form, performing exact inference using the sum-product algorithm (Tarlow et al. 2012) would be convenient. The basic idea has been presented in (Gail, Lubin, and Rubinstein 1981)



(a) (b)

Figure 1: The number of activated hidden units of CaRBM (a), and naive GC-RBM (b) on MNIST

Gaussian Cardinality Restricted Boltzmann Machines

While CaRBM shows its advantage in several experimental results, its assumption that there are no more than k hidden units could be activated simultaneously is too strict for many real world applications. So it should be more reasonable to find a certain kind of distribution to model the thresholds of the number of activated hidden units for different kinds of data, rather than to use a universal threshold.

In this paper, we propose a model which is capable of learning the thresholds by combining the user specified prior knowledge and the posterior knowledge after examining data. The learning algorithm framework shows the user specified term, which is the probability distribution function (pdf) of a certain distribution, can be replaced by the pdf of another distribution.

In this work, we choose Gaussian distribution as the user specified prior knowledge with the following reasons: (1) The statistics of our experiments on standard RBM suggest that the number of activated hidden units approximately follows Gaussian distribution. Our experimental results also show that the model using Gaussian distribution works well. (2) We also know that the number of activated hidden units obeys binomial distribution if all conditional probabilities $P(h_i = 1 \mid \mathbf{v})$ are the same. Although they are not totally the same, we still can consider the number of activated hidden units obeys a binomial-like distribution. However, binomial distribution is not able to model the variance of data, since there is only one parameter in binomial distribution after the number of hidden units is decided. (3) The number of parameters in the binomial-like distribution, which is consistent with N_h, is too many and difficult to handle with. Therefore, we need a simplified form to specify the number of activated hidden units. (4) According to the central limit theorem, the binomial distribution could be approximated by Gaussian when N_h is large enough and $P(h_i = 1 \mid \mathbf{v})$ is small enough.

Since we use Gaussian distribution as user specified prior

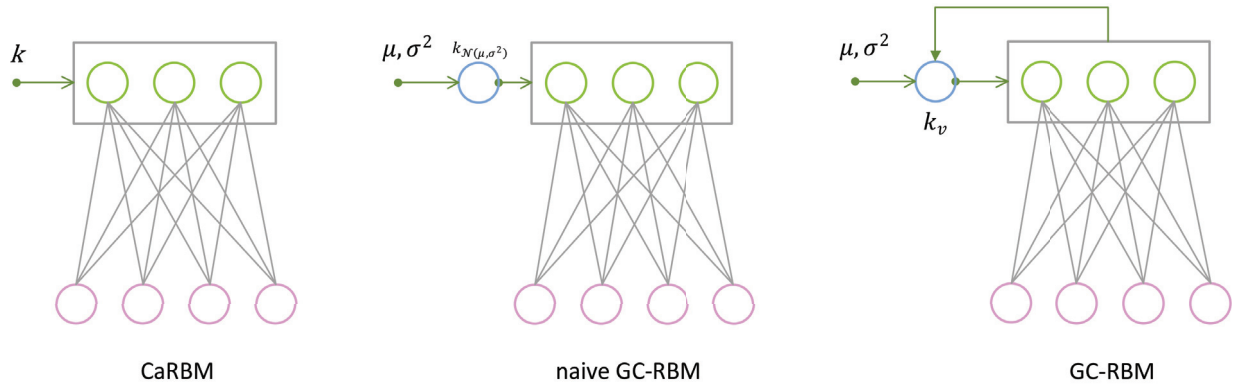


Figure 2: The comparison of CaRBM, naive GC-RBM and GC-RBM

to model the threshold of the number of activated hidden units, we call this model GC-RBM. The cardinality potential in GC-RBM is called Gaussian cardinality potential, which is quite similar with the cardinality potential of CaRBM. The joint probability of GC-RBM is given by:

$$P(\mathbf{v},\mathbf{h} \mid k_{\mathcal{N}(\mu,\sigma^2)}) = \frac{1}{Z} exp(\mathbf{v}^\top W\mathbf{h} + \mathbf{v}\mathbf{b} + \mathbf{h}^\top \mathbf{c}) \cdot \psi_{k_{\mathcal{N}(\mu,\sigma^2)}} \left(\sum_{j=1}^{N_h} h_j \right) \tag{5}$$

Given an input image $\mathbf{v}$, $k_{\mathcal{N}(\mu,\sigma^2)}$ is a sample drawn from a given Gaussian distribution $\mathcal{N}(\mu,\sigma^2)$ where μ is mean and σ is standard deviation. We also set a lower bound (e.g. 10) in order to filter all the negative values when sample $k_{\mathcal{N}(\mu,\sigma^2)}$. In practise, μ is usually not so small, resulting in that there is little chance to sample negative values. We would refer to it as *naive GC-RBM* the model defined by Eq.5 in order to distinguish it with GC-RBM which we elaborate soon. Note that, naive GC-RBM could also be seen as training a CaRBM for each input image, of course, each model have different cardinality potential, but all the models share their weights. The work (Hinton and Salakhutdinov 2009) also takes this view to help understand their model.

Figure 1 demonstrates the number of activated hidden units of both CaRBM and naive GC-RBM, where the k in CaRBM is 20 and the μ in naive GC-RBM is also 20. What we want is that the threshold of hidden unit activations varying with the actual data. In the case of Figure 1, the image which are relatively more complicated can be modeled with the threshold greater than 20. In this way, all the data should be modeled more precisely. However, we are not able to know the true threshold of a given image and the approach to obtain thresholds in naive GC-RBM absolutely relies on prior knowledge of humans, which means it ignores the input data. So we propose GC-RBM which is capable of adapting the thresholds of the number of activated hidden units to the input data, given by Eq.6:

$$P(\mathbf{v},\mathbf{h} \mid k_\mathbf{v}) = \frac{1}{Z} exp(\mathbf{v}^\top W\mathbf{h} + \mathbf{v}\mathbf{b} + \mathbf{h}^\top \mathbf{c}) \cdot \psi_{k_\mathbf{v}} \left(\sum_{j=1}^{N_h} h_j \right), \tag{6}$$

where $k_\mathbf{v}$ is the threshold of the number of activated hidden units. Given input data $\mathbf{v}$. $k_\mathbf{v}$ could be sampled from:

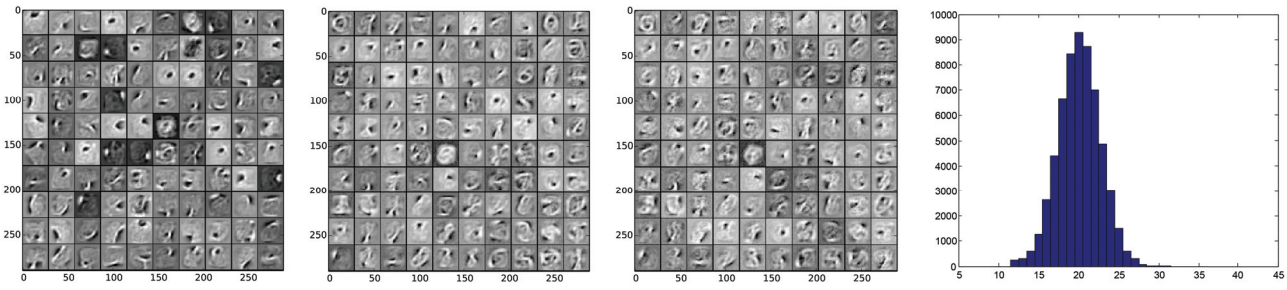
$$P(k_\mathbf{v} \mid \theta, \mathbf{v}) = \frac{P(\theta, \mathbf{v} \mid k_\mathbf{v}) P(k_\mathbf{v})}{P(\theta, \mathbf{v})}. \tag{7}$$

where θ represents the parameters specified by users.

$$\begin{aligned} P(k_\mathbf{v} \mid \mu, \sigma^2, \mathbf{v}) &= \frac{P(\theta \mid k_\mathbf{v}) P(\mathbf{v} \mid k_\mathbf{v}) P(k_\mathbf{v})}{P(\theta) P(\mathbf{v})} \\ &= \frac{P(\theta \mid k_\mathbf{v}) P(k_\mathbf{v})}{P(\theta)} \cdot \frac{P(\mathbf{v} \mid k_\mathbf{v}) P(k_\mathbf{v})}{P(\mathbf{v})} \cdot \frac{1}{P(k_\mathbf{v})} \\ &= P(k_\mathbf{v} \mid \theta) \cdot P(k_\mathbf{v} \mid \mathbf{v}) \cdot \frac{1}{P(k_\mathbf{v})} \end{aligned} \tag{8}$$

From Eq.8, $P(k_\mathbf{v} \mid \theta, \mathbf{v})$ can be obtained by three parts: (1) $P(k \mid \theta)$, representing the user specified prior knowledge on the thresholds of the number of activated hidden units, is replaced by μ and σ^2 in GC-RBM. (2) $P(k_\mathbf{v} \mid \mathbf{v})$, representing the posterior knowledge on k after examining the data, is the conditional probability given data $\mathbf{v}$. Related inference is similar to that in CaRBM. But the time complexity of computing exact values of $P(k_\mathbf{v} \mid \mathbf{v})$ would not be tolerated since we need to compute that for every input data. So here we assume the $P(k_\mathbf{v} \mid \mathbf{v}) \sim \mathcal{N}(\sum_i^{N_h} P(h_i = 1 \mid \mathbf{v}), \sigma^2)$. The reason for this assumption is explained earlier in this paper. Based on this assumption, we can sample $P(k_\mathbf{v} \mid \theta, \mathbf{v})$ by MCMC. (3) $P(k_\mathbf{v})$, is the prior of $k_\mathbf{v}$, of which the ground truth we can not obtained. Here we assume $P(k_\mathbf{v})$ follows uniform distribution, empirically. Algorithm1 is the learning algorithm of GC-RBM. Details of updating **W,b,c** can be referred to the work (Swersky et al. 2012).

Eq.8 indicates that with high probability, we sample thresholds of which values are between two means of two



(a) (b) (c) (d)

Figure 3: (a),(b),(c)Visualization of the weights of 100 hidden units in CaRBM, naive GC-RBM and GC-RBM models on MNIST. (d) The histogram of the number of activated hidden units in GC-RBM on MNIST.

Gaussian distributions, and the chance to sample values far from two means is low, as well. It means that we learn thresholds of which the probabilities are regarded as weights. Probabilities between two means were initialized with relatively higher values. Then the high values zone keep approaching the position where the mean of given Gaussian distribution as the consequences of increasing of n_k.

Algorithm 1 Learning Algorithm of GC-RBM on Pretraining Phase

Input:
(1) Training data, D.
(2) Parameters of Gaussian prior, μ, σ^2.
(3) Number of iterations for updating $k_{\mathbf{v}}$, n_k.
(4) Number of iterations for updating $\mathbf{W}, \mathbf{b}, \mathbf{c}$, n_p.
Output:
The parameters of GC-RBM, $\mathbf{W}, \mathbf{b}, \mathbf{c}$.

for each training image $\mathbf{v}$ in D **do**
  Sample $k_{\mathbf{v}} \sim \mathcal{N}(\mu, \sigma^2)$;
  for $i \leftarrow 1$ to n_k **do**
    Randomly initialize $\mathbf{W}, \mathbf{b}, \mathbf{c}$;
    for $j \leftarrow 1$ to n_p **do**
      Updating $\mathbf{W}, \mathbf{b}, \mathbf{c}$ with $k_{\mathbf{v}}$ as CaRBM;
    end for
    if $i < n_k$ **then**
      Calculate the value of $P(k_{\mathbf{v}} \mid \mu, \sigma^2, \mathbf{v})$;
      Sample $k_{\mathbf{v}}$ from $P(k_{\mathbf{v}} \mid \mu, \sigma^2, \mathbf{v})$;
    end if
  end for
end for

Figure 2 shows the comparison of three models, which could help understand the differences among the three models. In a word, (1) CaRBM sets a uniform threshold of hidden unit activations for all input images; (2) Naive GC-RBM sets thresholds sampled from a given Gaussian distribution for different input data; (3) GC-RBM takes the feedback of hidden units. Actually, naive GC-RBM is not that different from CaRBM, since the sparse conditions in both two models are independent of input data. On the contrast, the thresholds of the number of activated hidden units in GC-RBM are able to adapt to input data.

Parameter Settings

There are three more parameters (μ, σ, and n_k) than ones in CaRBM. The n_p is assigned the same value as the number of training iterations in CaRBM. The most appropriate values of μ and σ depend on the data set. The influence of μ and σ will be demonstrated in the experiment section.

n_k is used to scale the outer loop. It indicates the degree of similarity between the given Gaussian distribution and the distribution of thresholds of the number of activated hidden units after learning, or an importance weight of user specified prior knowledge. The greater n_k is, the distribution after learning will be more like the given Gaussian distribution, which means the given Gaussian distribution plays a more important role in GC-RBM. In practice, large value of n_k is not necessary, which is 2 in our experiment. Settings of other parameters that are mentioned in our algorithm can be referred to the work (Swersky et al. 2012).

Complexity

We analyze the time complexity of Algorithm1 in this section. Obviously, for each input image $\mathbf{v}$, sampling $k_{\mathbf{v}}$ costs $O(1)$. Calculating $k_{\mathbf{v}}$ from $P(k_{\mathbf{v}} \mid \mu, \sigma^2, \mathbf{v})$ costs $O(N_h)$, since we need obtain the value of $\sum_i^{N_h} P(h_i = 1 \mid \mathbf{v})$. Sampling $k_{\mathbf{v}}$ costs $O(1)$ as well. It seems that the extra time cost by learning algorithm of GC-RBM is just $O(n_k \cdot N_h)$ and $n_k - 1$ times for updating $\mathbf{W}, \mathbf{b}, \mathbf{c}$. However, sampling $k_{\mathbf{v}}$ using MCMC procedure costs quite a lot of time. So in practice, we assign a relatively smaller value to n_p when $i = n_k$ and a larger value when $i < n_k$.

Experiment

The goals of our experiment are as follows: (1) Show the performance of GC-RBM and compare it with other two models of which the thresholds independent of the data. (2) Analyze parameters in GC-RBM which influence the performance of GC-RBM on classification tasks.

Datasets

The experiments were conducted on MNIST and CIFAR-10 (Krizhevsky and Hinton 2009). MNIST has been a benchmark data set for image classification task that contains 70000 28*28 grayscale images of handwritten dig-

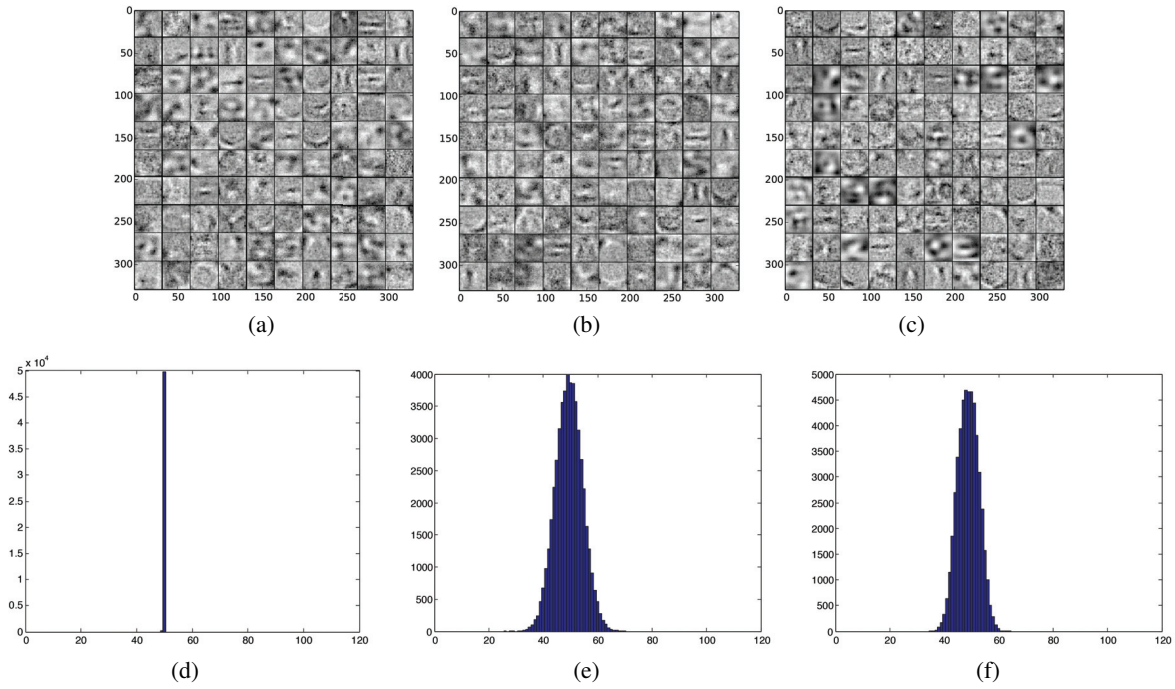


Figure 4: (a),(b),(c)Visualization of the weights of 100 hidden units in CaRBM, naive GC-RBM and GC-RBM on CIFAR-10. (d),(e),(f) The histogram of the number of activated hidden units in CaRBM, naive GC-RBM and GC-RBM on CIFAR-10.

its. 60000 images for training and 10000 images for test. Dataset CIFAR-10 consists of 60000 32*32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. Compared with images in CIFAR-10, those in MNIST are relatively simpler to be recognized.

Classification on MNIST

We applied the CaRBM, naive GC-RBM and GC-RBM to train a three layers (784-100-10) feed-forward neural network (the first layer is one of models mentioned above, the second layer is a softmax classifier) on MNIST. Similar to RBM, we pre-train the network in an unsupervised fashion and then fine-tune it in a supervised way. In pre-training phase, the settings of parameters which we do not mention in this paper followed the work (Swersky et al. 2012). The μ and σ in GC-RBM and naive GC-RBM were assigned to $\mu \in \{10, 20, ..., 100\}, \sigma^2 \in \{9, 25, 100\}$. In order to have comparisons among three models, the k in CaRBM was assigned to the same values as the μ in our Gaussian thresholds models. The settings of parameters on fine-tuning phase also followed the work (Swersky et al. 2012).

Another problem is how to decide thresholds for test data. In the case of naive GC-RBM, we sampled thresholds for test data from the Gaussian distribution of which parameters are the same as those on pre-training phase. In the case of GC-RBM, we also used the Algorithm1, except that we only update thresholds once, since the network has finished learning.

Figure 3(a), 3(b), 3(c) show the weights learnt by three models. Figure 3(d) is the thresholds of the number of activated hidden units in GC-RBM where the mean is 20. We used Gaussian distribution to fit both thresholds before and after learning. We find that fitted parameters before learning are $(\mu = 19.4986, \sigma = 3.0154)$ and those after learning are $(\mu = 20.0686, \sigma = 2.6176)$[1].

Parts of experimental results are showed in Figure 5. Other results are omitted since there is significant increase of classification error with k, μ from 50 to 90. This issue occurred in our experiments on CIFAR-10 as well. Each result is the average of the results of 10 runs with the same parameter settings. From Figure 5 we can see that: (1) GC-RBM shows better performance than other models and obtains the lowest error with $\mu = 20$ and $\sigma = 5$. We calculated the P value with the parameters $k = \mu = 20, \sigma = 5$ on MNIST, and the P value is less than 0.0001, which means the difference is extremely statistically significant. (2) The classification error of CaRBM where $k = 10$ is the highest. The results might be due to the strict constraint of CaRBM while other models have chance to sample thresholds larger than 10, with which the model is more powerful in terms of representing data. (3) The performance of GC-RBM where $k, \mu = 40$ is poorer than CaRBM. The reason for this might be that GC-RBM modeled more redundancy where $k, \mu = 40$ (4) The results also suggest that the variance should be assigned to a relatively smaller value, since both naive GC-RBM and GC-RBM demonstrate poor performance where $\sigma = 10$.

[1]The thresholds after learning do not obey Gaussian distribution, here we just use the numbers to represent the difference.

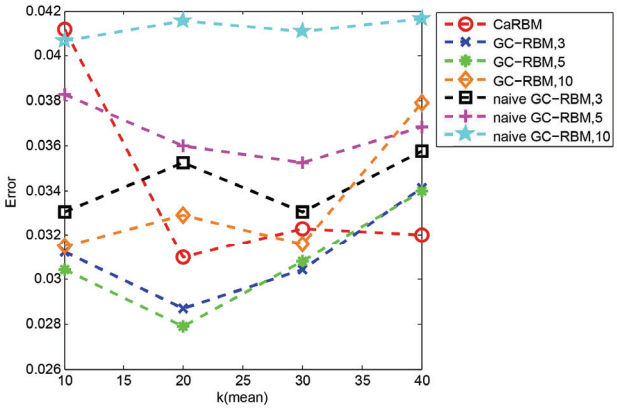


Figure 5: Classification performance on MNIST

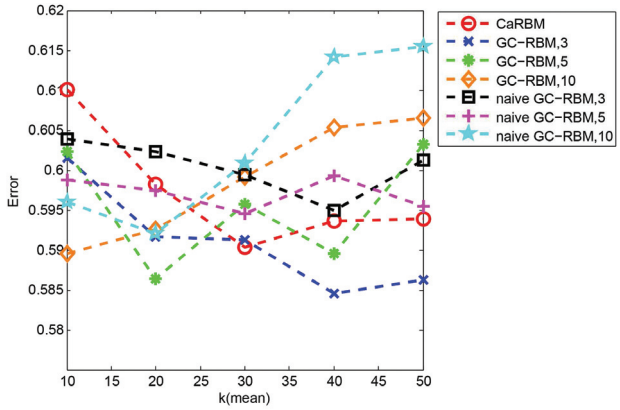


Figure 6: Classification performance on CIFAR-10

Classification on CIFAR-10

Since CIFAR-10 is a color image data set, we whitened all the images and trained the three models. Except the difference in the amount of visible units on two data sets, other parameter settings are the same as those on MNIST. Figure 4(a), 4(b), 4(c) show the weights learnt by three models on CIFAR-10. Figure 4(d), 4(e), 4(f) show the number of activated hidden units of three models.

Figure 6 shows the performance of three models on CIFAR-10. What we can read from the figure is as follows: (1) GC-RBM obtains the lowest classification errors with $\mu = 40$ and $\sigma = 3$. The P value with parameters $k = \mu = 40, \sigma = 3$ on CIFAR-10 is also less than 0.0001. The reason for the larger values of k and μ where the lowest classification error is obtained, compared with which on MNIST, might be that images in CIFAR-10 are more complex than those in MNIST. (2) We also find that the classification error of CaRBM is the highest where $k, \mu = 10$. The reason might be the same as that on MNIST.

Comparing GC-RBM with CaRBM

Compared with the performance of naive GC-RBM on MNIST and CIFAR-10 dataset, the strict sparsity constraint is a advantage of CaRBM. However, it becomes a disadvantage when compared with GC-RBM.

From the experiment results, we found that it is important to adapt thresholds to input data, which GC-RBM is capable of. What is more important is that our learning framework can replace the Gaussian distribution by another distribution as user specified prior knowledge. Utilizing Gaussian distribution is easy to implement since there are only two more parameters than CaRBM and it works well in our experiments.

In our experiments, GC-RBM is able to obtain the lower classification error than CaRBM, while the opposite results might be found if the parameter settings are far from the optimum, such as the results with $k, \mu = 40$ on MNIST.

Conclusion

We proposed a new model on the basis of CaRBM by replacing the universal threshold of hidden unit activations for all input data with thresholds for different input data. We proposed a principled algorithm to learn the thresholds by given Gaussian distribution and training data, and showed how other distribution can be used instead in our algorithm frameworks. We obtained better experimental results compared to CaRBM on MNIST and CIFAR-10. We also analyzed how the parameters in GC-RBM influence the performance on classification task.

Acknowledgements

The work is supported by National Basic Research Program of China (2015CB352300) and National Natural Science Foundation of China (61271394).

References

Bengio, Y.; Lamblin, P.; Popovici, D.; and Larochelle, H. 2007. Greedy layer-wise training of deep networks. *Advances in neural information processing systems* 19:153.

Bengio, Y. 2009. Learning deep architectures for ai. *Foundations and trends in Machine Learning* 2(1):1–127.

Cho, K.; Raiko, T.; and Ilin, A. 2011. Gaussian-bernoulli deep boltzmann machine. In *NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning*.

Gail, M. H.; Lubin, J. H.; and Rubinstein, L. V. 1981. Likelihood calculations for matched case-control studies and survival studies with tied death times. *Biometrika* 68(3):703–707.

Goh, H.; Thome, N.; Cord, M.; et al. 2010. Biasing restricted boltzmann machines to manipulate latent selectivity and sparsity. In *NIPS workshop on deep learning and unsupervised feature learning*.

Goh, H.; Kusmierz, L.; Lim, J.-H.; Thome, N.; and Cord, M. 2011. Learning invariant color features with sparse topographic restricted boltzmann machines. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 1241–1244. IEEE.

Gupta, R.; Diwan, A. A.; and Sarawagi, S. 2007. Efficient inference with cardinality-based clique potentials. In *Proceedings of the 24th international conference on Machine learning*, 329–336. ACM.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507.

Hinton, G. E., and Salakhutdinov, R. 2009. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, 1607–1614.

Hinton, G. E.; Osindero, S.; and Teh, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18(7):1527–1554.

Hinton, G. E. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation* 14(8):1771–1800.

Hinton, G. 2010. A practical guide to training restricted boltzmann machines. *Momentum* 9(1).

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*.

Lee, H.; Grosse, R.; Ranganath, R.; and Ng, A. Y. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 609–616. ACM.

Lee, H.; Ekanadham, C.; and Ng, A. 2007. Sparse deep belief net model for visual area v2. In *Advances in neural information processing systems*, 873–880.

Luo, H.; Shen, R.; Niu, C.; and Ullrich, C. 2011. Sparse group restricted boltzmann machines. In *AAAI*.

Salakhutdinov, R., and Hinton, G. E. 2009. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, 448–455.

Snoek, J.; Adams, R. P.; and Larochelle, H. 2012. Nonparametric guidance of autoencoder representations using label information. *Journal of Machine Learning Research* 13:2567–2588.

Swersky, K.; Sutskever, I.; Tarlow, D.; Zemel, R. S.; Salakhutdinov, R.; and Adams, R. P. 2012. Cardinality restricted boltzmann machines. In *Advances in Neural Information Processing Systems*, 3302–3310.

Tarlow, D.; Swersky, K.; Zemel, R. S.; Adams, R. P.; and Frey, B. J. 2012. Fast exact inference for recursive cardinality models. *arXiv preprint arXiv:1210.4899*.

Tarlow, D.; Givoni, I. E.; and Zemel, R. S. 2010. Hop-map: Efficient message passing with high order potentials. In *International Conference on Artificial Intelligence and Statistics*, 812–819.