Sparse Group Restricted Boltzmann Machines

Heng Luo† Ruimin Shen† Changyong Niu‡ Carsten Ullrich†
†Shanghai Jiao Tong University
‡Zhengzhou University
†{hengluo, rmshen, ullrich_c}@sjtu.edu.cn ‡iecyniu@zzu.edu.cn

Abstract

Since learning in Boltzmann machines is typically quite slow, there is a need to restrict connections within hidden layers. However, the resulting states of hidden units exhibit statistical dependencies. Based on this observation, we propose using l_1/l_2 regularization upon the activation probabilities of hidden units in restricted Boltzmann machines to capture the local dependencies among hidden units. This regularization not only encourages hidden units of many groups to be inactive given observed data but also makes hidden units within a group compete with each other for modeling observed data. Thus, the l_1/l_2 regularization on RBMs yields sparsity at both the group and the hidden unit levels. We call RBMs trained with the regularizer *sparse group* RBMs (SGRBMs). The proposed SGRBMs are applied to model patches of natural images, handwritten digits and OCR English letters. Then to emphasize that SGRBMs can learn more discriminative features we applied SGRBMs to pretrain deep networks for classification tasks. Furthermore, we illustrate the regularizer can also be applied to deep Boltzmann machines, which lead to sparse group deep Boltzmann machines. When adapted to the MNIST data set, a two-layer sparse group Boltzmann machine achieves an error rate of 0.84%, which is, to our knowledge, the best published result on the permutation-invariant version of the MNIST task.

Introduction

Restricted Boltzmann Machines (RBMs) (Smolensky 1986; Freund and Haussler 1994; Hinton 2002) recently have become very popular because of their excellent ability of unspervised learning, and have been successfully applied in various application domains, such as dimensionality reduction (Hinton and Salakhutdinov 2006), Object Recognition (Lee et al. 2009) and others.

For the purpose of obtaining efficient and exact inference, there are no connections within the hidden layer in RBMs. But by considering statistical dependencies of states of hidden units we may learn a more powerful generative model (Garrigues and Olshausen 2008). Following this idea, in order to consider, at least to some extent, statistical dependencies of states of hidden units and meanwhile keeping the exact and efficient inference in RBMs, we introduce a l_1/l_2 regularizer on the activation probabilities of hidden units given training data.

l_1/l_2 regularizer has been intensively studied in both the statistics community (Yuan and Lin 2006) and machine learning community (Bach 2008). Usually the l_1/l_2 regularizer (or group lasso) only leads to sparsity at the group level but not within a group. In this paper, we show that introducing the l_1/l_2 regularizer on the activation probabilities can yield sparsity at not only the group level but also the hidden unit level. Thus, we call RBMs trained with the regularizer sparse group RBMs (SGRBMs). Empirically we show that SGRBMs can achieve better generative and discriminative performances than RBMs. Furthermore, we also show the regularizer can be easily applied to train deep Boltzmann Machines.

Restricted Bolzmann Machines and Contrastive Divergence

An RBM is a two layer neural network with one visible layer representing observed data and one hidden layer as feature detectors. Connections only exist between the visible layer and the hidden layer. Here we assume that both the visible and hidden units of the RBM are binary. The models below can be easily generalized to other types of units (Welling, Rosen-Zvi, and Hinton 2005). The energy function of an RBM is defined as

$$E(x,h) = -\sum_{i,j} x_i h_j w_{ij} \tag{1}$$

where x_i and h_j denote the states of the ith visible unit and the jth hidden unit, while w_{ij} represents the strength of the connection between them. For simplicity, we omit the biases of the visible and hidden units.

Based on the energy function, we can define the joint distribution of (x,h)

$$P(x,h) = \frac{1}{Z}\exp(-E(x,h)) \tag{2}$$

$$Z = \sum_{\widetilde{x},\widetilde{h}} \exp(-E(\widetilde{x},\widetilde{h})) \tag{3}$$

where Z is the partition function.

Given observed data, the states of the hidden units are conditionally independent. Their activation probabilities are,

$$P(h_j|x) = \frac{1}{1+\exp(-x^T w_{.j})} \tag{4}$$

where $w_{.j}$ denotes the jth column of W, which is the connection weights between the jth hidden unit and all visible units. If more data in the training data set can activate a hidden unit with a high probability, the hidden unit's feature will be less discriminative. Thus it is sometimes necessary to introduce sparsity in the hidden layer of an RBM (Lee, Ekanadham, and Ng 2008; Larochelle and Bengio 2008; Salakhutdinov and Larochelle 2010).

The marginal distribution over the visible units actually is a model of products of experts (Hinton 2002)

$$\begin{aligned} P(x) &= \frac{1}{Z}\prod_j (1+\exp(x^T w_{.j})) \\ &= \frac{1}{Z}\prod_j 1/(1-P(h_j=1|x)) \end{aligned} \tag{5}$$

From Equation 5 we can deduce that each expert (hidden unit) will contribute probabilities according to the activation probability given the data vector x. If given a data sample one specific hidden unit will be activated with a high probability, we say the hidden unit is responsible for *representing* the data sample.

The objective of generative training of an RBM is to model the marginal distribution of the visible units $P(x)$. To do this, we need to compute the following gradient given the training data $x^{(l)}$

$$-<\frac{\partial E(x^{(l)},h)}{\partial\theta}>_{P(h|x^{(l)})} + <\frac{\partial E(x,h)}{\partial\theta}>_{P(x,h)} \tag{6}$$

where $<\ .\ >_P$ is the expectation with respect to the distribution P. The second term of Equation 6 is intractable since sampling the distribution $P(x,h)$ requires prolonged Gibbs sampling. Hinton (2002) shows that we can get very good approximations to the second term when running the Gibbs sampler only k steps, initialized from the training data. Named Contrastive Divergence (CD) (Hinton 2002), the algorithm updates the feature of the jth hidden unit seeing the training data $x^{(l)}$

$$\Delta w_{.j} = P(h_j=1|x^{(l)})\cdot x^{(l)} - P(h_j=1|x^{(l)-})\cdot x^{(l)-} \tag{7}$$

where $x^{(l)-}$ is sampled from $P(x|h^{(l)})$ ($h^{(l)}$ sampled from $P(h|x^{(l)})$). The first term of Equation 7 will decrease the energy of $x^{(l)}$ (which cause that $x^{(l)}$ would be more probable under the RBM) (Bengio 2009). At the same time this term also guarantees that hidden unit j will be activated with a higher probability when the hidden unit see $x^{(l)}$ again, which means hidden unit j are *learning to represent* $x^{(l)}$. Because the hidden states of different hidden units are conditionally independent given the data, all of hidden units will independently learn to represent $x^{(l)}$ with different speeds which are decided by their own activation probabilitiy ($P(h_j|x^{(l)})$). The learning process does not stop until the reconstruction is perfect ($x^{(l)} = x^{(l)-}$). In the next section, we introduce a mixed-norm (l_1/l_2) regularizer on the activation probabilities of hidden units given the training data to make sure the learning process are not conditionally independent and encourage the sparsity in the hidden units.

Sparse Group RBMs

Directly learning the statistical dependencies between all of hidden units is inefficient. To alleviate this problem, firstly we equally divide hidden units into predefined non-overlapping groups to restrain the dependencies within these groups. Secondly, instead of learning the dependencies we penalize the overall activation level of a group. To implement the two above intuitions we introduce a mixed-norm (l_1/l_2) regularizer on the activation probabilities of hidden units given the training data.

Assuming an RBM has F hidden units, let $\mathcal{H}$ denote the set of all hidden units' indices: $\mathcal{H} = \{1,2,...,F\}$. The kth group is denoted by $\mathcal{G}_k$ where $\mathcal{G}_k \subset \mathcal{H}, k = 1,...,K$. In this paper, we set all of groups being non-overlapping and with equal size. Given a grouping $\mathcal{G}$ and a data sample $x^{(l)}$, the kth group norm $N_k(x^{(l)})$ is given by

$$N_k(x^{(l)}) = \sqrt{\sum_{m\in G_k} P(h_m=1|x^{(l)})^2} \tag{8}$$

which is the Euclidean (l_2) norm of the vector composed of these activation probabilities and considered as the overall activation level of kth group. Given all the group norms, the mixed-norm is

$$\sum_{k=1}^{K}|N_k(x^{(l)})| = \sum_{k=1}^{K}\sqrt{\sum_{m\in G_k} P(h_m=1|x^{(l)})^2} \tag{9}$$

which is the l_1 norm of the vector composed of the group norms.

We add the l_1/l_2 regularizer to the log-likelihood of training data. Thus, given training data, we need to solve the following optimization problem

$$\max_{W,b,c}\sum_{l=1}^{L}\log P(x^{(l)}) - \lambda\sum_{k=1}^{K}N_k(x^{(l)}) \tag{10}$$

where λ is a regularization constant. To solve Equation 10, we can apply the contrastive divergence update rule (see Equation 6), followed by one step of gradient ascent using the gradient of the regularization term.

The effects of a l_1/l_2 regularization can be interpreted on two levels: an across-group and a within-group level. On the across-group level, the group norms N_k behave as if they were penalized by a l_1 norm. In consequence, given observed data some group norms are zero, which means the activation probabilities of all hidden units in these groups are zero since the activation probabilities are non-negative. In other words, given a data sample only few groups' hidden units are responsible to represent it. On the within-group level, the l_2 norm will equally penalize the activation probabilities of all hidden units in the same group. The l_2

norm thus does not yield sparsity within the group. However, when applied the l_1/l_2 on the activation probabilities it is an entirely different story because of the Logistic differential equation. Below we will discuss it in detail.

By introducing this regularizer, Equation 7 is changed to the following equation

$$\Delta w_{\cdot j} = P(h_j = 1|x^{(l)}) \cdot x^{(l)} - P(h_j = 1|x^{(l)}) \cdot x^{(l)-} - \lambda \cdot \alpha \tag{11}$$

$$\begin{aligned} \alpha &= \frac{\partial}{\partial w_{\cdot j}} \sum_{k=1}^{K} N_k(x^{(l)}) \\ &= \frac{P(h_j = 1|x^{(l)})}{N_{k'}(x^{(l)})} \cdot \frac{\partial}{\partial w_{\cdot j}} P(h_j = 1|x^{(l)}) \\ &= \frac{1}{N_{k'}(x^{(l)})} P(h_j = 1|x^{(l)})^2 P(h_j = 0|x^{(l)}) \cdot x^{(l)} \end{aligned} \tag{12}$$

where we assume the jth hidden unit belongs to the k'th group. The last step in Equation 12 uses the fact that $\frac{d}{dt}P(t) = P(t)(1 - P(t))$ if P is a logistic function. Unlike in Equation 7, hidden unit j is not independently learning to represent $x^{(l)}$. The learning is now determined by the activation probability $P(h_j = 1|x^{(l)})$ but also the overall activation level of other hidden units in the same group, $\sum_{m \in G_{k'}, m \neq j} P(h_m = 1|x^{(l)})^2$. More specifically, the speed of learning from $x^{(l)}$ is slowed by the following factor,

$$\frac{P(h_j = 1|x^{(l)})^2 P(h_j = 0|x^{(l)})}{\sqrt{\sum_{m \in G_{k'}, m \neq j} P(h_m = 1|x^{(l)})^2 + P(h_j = 1|x^{(l)})^2}} \tag{13}$$

We visualize the factor in Figure 1 where we assume there are 5 hidden units in the group and the overall activation levels are thus in the interval $(0, 4)$.

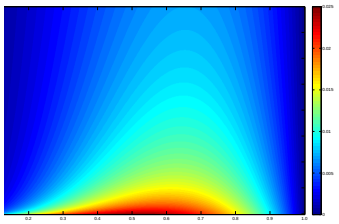


Figure 1: The factor defined in Equation 13. Here the horizontal axis represents the activation probability $P(h_j = 1|x^{(l)})$ and the vertical axis represents the overall activation level of other hidden units in the same group, $\sum_{m \in G_{k'}, m \neq j} P(h_m = 1|x^{(l)})^2$.

In Figure 1, when the other hidden units in the group are inactive for the data $x^{(l)}$ (a small value of $\sum_{m \in G_{k'}, m \neq j} P(h_m = 1|x^{(l)})^2$) the jth hidden unit is penalized strongly. In other words, hidden unit j is prohibited from learning from $x^{(l)}$ because of the low group activation level. The first property of the l_1/l_2 regularizer is that it encourages few groups to be active given observed data. This property yields the sparsity at the group level. In Figure 1 it can also be seen that the effect of the regularizer will vanish when $P(h_j = 1|x^{(l)})$ is close to 0 or 1. More specifically, the effects of the regularizer diminish more quickly when the activation probabilities are close to 0 instead of to 1 because of the square of $P(h_j = 1|x^{(l)})$ in Equation 13. It can be interpreted as that hidden units in a group compete with each other for learning to represent the data sample $x^{(l)}$ (When $x^{(l)} = x^{(l)-}$ the competition stops). Usually few of hidden units in a group will win this competition. Thus, the second property of the regularizer is that it results in only a few hidden units to be active in a group. This property yields the sparsity within the group. Based on these two properties, we call RBMs trained by Equation 10 sparse group RBMs (SGRBMs).

Relationship to third-order RBMs

A third-order RBM can be formed as a mixture model whose components are RBMs (Nair and Hinton 2009). To a certain extent, a trained third-RBM defines a special **group sparse** representation for training data. Discussing the relationships between a third-RBM and sparse group RBMs will give us additional insights about the effects of l_1/l_2 regularizer for RBMs.

The energy function of a third-order Boltzmann machine is

$$E(x, h, z) = -\sum_{i,j,k} x_i h_j^k w_{ij}^k z_k \tag{14}$$

where z is a K-dimensional binary vector with 1-of-K activation and represents the cluster label. The responsibility of the kth component RBM is

$$\begin{aligned} P(z_k = 1|x) &= \frac{\sum_h P(x, z_k = 1, h)}{\sum_{l=1}^{K} P(x, z_l = 1)} \\ &= \frac{\prod_j (1/(1 - P(h_j^k = 1|x))}{\sum_{l=1}^{K} \prod_{j'} (1/(1 - P(h_{j'}^l = 1|x))} \end{aligned} \tag{15}$$

A third-order RBM with K components can be seen as a regular RBM in which hidden units are divided into K non-overlapping groups. Given a data x the responsibility, $P(z|x)$ is used to pick one group's hidden units to respond to the data. In other words, data will be represented by only one group's hidden units and the states of hidden units in other groups will be set to 0. From this perspective, a third-order RBM yields a special group sparsity given the training data.

The group (component) which has the bigger value of $\prod_j (1/(1 - P(h_j^k = 1|x))$ will more likely be responsible for the data x. Given the data x the product $\prod_j (1/(1 - P(h_j^k = 1|x))$ can be interpreted as a measure of overall activation level of hidden units in the group. If more hidden units in the group are active, the overall activation level of the group is higher. However the products are unbounded and at very different numerical scales since any hidden unit's activation probability ($P(h_j^k = 1|x)$) in a group that is close to 1 will make the product extremely big. To alleviate this problem,

Nair and Hinton (2009) introduced a temperature parameter T to reduce scale differences in the products.

There are two major differences between third-order RBMs and SGRBMs. Firstly, SGRBMs define a different overall activation level of a group's hidden units, which is the euclidean norm of the vector, $(P(h_j = 1|x))_{j\in\mathcal{G}_k}$. Since this measure is bounded and in the interval $(0, |\mathcal{G}_k|)$, it can be avoided that one group with a too high overall activation level shields all of other groups. Secondly, as discussed in the previous section, SGRBMs yields sparsity at both the group level and the hidden unit level by regularization.

Sparse Group Deep Bolzmann Machines

Salakhutdinov and Hinton (2009) presented an algorithm for tractable training multilayer Bolzmann machines, in which, unlike deep belief networks, hidden units will receive top-down feedback. Unfortunately to keep inference efficient, there are still no connections within the hidden layer in the deep Bolzmann machines. We show that the proposed l_1/l_2 regularizer can also be easily added to DBMs. This leads to sparse group DBMs which can achieve better discriminative performances (see the experiment section).

Taking a two-layer Boltzmann machine for example, the energy function is

$$E(x, h^1, h^2) = -x^T W^1 h^1 - x^T W^2 h^2 \tag{16}$$

For training a sparse group deep Bolzmann machine (SGDBM), we propose the following optimization problem

$$\max_{W^1,W^2} \sum_{l=1}^{L} \log P(x^{(l)}) - \lambda_1 \sum_{k=1}^{K} N_k^1(x^{(l)}) - \lambda_2 \sum_{m=1}^{M} N_m^2(x^{(l)}) \tag{17}$$

$$N_k^1(x^{(l)}) = \sqrt{\sum_{j\in\mathcal{G}_k^1} P(h_j^1 = 1|x^{(l)})^2} \tag{18}$$

$$N_m^2(x^{(l)}) = \sqrt{\sum_{j\in\mathcal{G}_m^2} P(h_j^2 = 1|x^{(l)})^2}; \tag{19}$$

Given observed data the two activation probabilities can not be computed efficiently. Following Salakhutdinov and Hinton (2009), we used $P(h_j^1 = 1|x^{(l)}, \tilde{h^2})$ and $P(h_j^2 = 1|\tilde{h^1})$ to approximate $P(h_j^1 = 1|x^{(l)})$ and $P(h_j^2 = 1|x^{(l)})$ where $\tilde{h^2}$ and $\tilde{h^1}$ are the corresponding mean-field approximations.

Experiments

Since SGRBMs yields sparsity at the hidden units level, we firstly applied SGRBMs to model patches of natural images. We show that SGRBMs are able to learn localized, oriented, gabor-like features. Then to quantitatively evaluate the performances of SGRBMs (as generative models), we applied SGRBMs to the MNIST handwritten digit dataset[1] and the OCR English letters dataset[2] and reported the average test log-probabilities. For evaluating the discriminative performances of SGRBMs, we applied SGRBMs to pretain multilayer feedforward networks on the MNIST dataset and the OCR English letters dataset. At last we also trained SGDBMs on these two datasets.

Modeling Patches of Natural Images

The training data used consists of $100,000$ 14×14 patches randomly extracted from a standard set of 10 512×512 whitened images as in (Olshausen and others 1996). We divided all patches into mini-batches, each of which contained 200 patches, and updated the weights after each mini-batch.

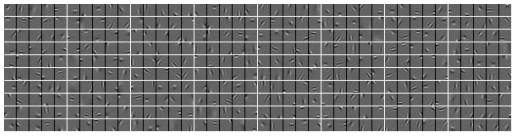


Figure 2: Learned features with the SGRBM trained on patches of natural images. The white lines denote the boundaries of the groups.

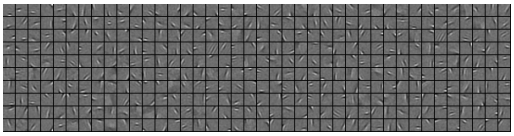


Figure 3: Learned features with sparse RBM trained on patches of natural images

We trained a SGRBM with 196 real-valued visible units and 400 hidden units which are divided into 80 uniform non-overlapping groups. There are 5 hidden units in each group. The regularization constant, λ (see Equation 10), is empirically set to 0.1[3]. The learned features are shown in Figure 2. For comparison, we also trained a sparse RBM (Lee, Ekanadham, and Ng 2008) with 400 hidden units. The learned features are shown in Figure 3. Since hidden units in a group compete with each other to model pathes, each hidden unit in the SGRBM is focused on modeling more subtle patterns contained in training data. As a result, the features learned with the SGRBM are more localized than those learned with the sparse RBM.

Modeling Handwritten Digits and OCR English Letters

The MNIST digit dataset contains $60,000$ training and $10,000$ test 28×28 images. We further randomly split the training set into $50,000$ training and $10,000$ validation images[4]. OCR letters dataset contains $32,152$ 16×8 images. Following the code[5] provided by Larochelle, we split the

[1] http://yann.lecun.com/exdb/mnist/.

[2] http://ai.stanford.edu/~btaskar/ocr/.

[3] The results we report below were insensitive to the choice of λ.

[4] To make the comparison with previous results on the MNIST dataset fair, once good hyper-parameter values were selected based on the validation set, all 60,000 training examples were used to train the final model.

[5] http://www.cs.toronto.edu/~larocheh.

Table 1: The estimates of the variational lower bound on the average test log-probabilities.

Models	MNIST	OCR letters
RBM CD-1	-113.0	-33.9
SGRBM CD-1	-111.7	-32.7
RBM CD-25	-86.2	-29.0
SGRBM CD-25	-85.3	-28.7

dataset into $32,152$ training, $10,000$ validation and $10,000$ test examples. Both the MNIST training set and the OCR letters training set are divided into mini-batches, each of which contained 100 images.

We implement Model selection with a grid search over the learning rate (0.0001, 0.001, 0.01 or 0.1), the number of hidden units (250, 500, 750 and 1000), the group size for SGRBMs (5, 10, 20 and 50) and the regularization constant, λ for SGRBMs (0.001, 0.01, 0.1 and 1). We configure a SGRBMs with 500 hidden units and group size 5 for the MNIST dataset and a SGRBMs with 1000 hidden units and group size 5 for the OCR letters dataset. The regularization constant, λ is set to 0.1 for both of the datasets. We compare these models to two regular RBMs with 500 and 1000 hidden units, respectively. All of these models are trained using CD with $k = 1$ and $k = 25$, respectively.

Although computing the exact partition function of an RBM is intractable, Salakhutdinov and Murray (2008) proposed an Annealed Importance Sampling based algorithm to tractably approximate the partition function of an RBM. Using their method, the estimates of the lower bound on the average test log-probabilities are given in Table 1. It can be seen that by adopting the l_1/l_2 regularization we can learn better generative models on the MNIST dataset and the OCR dataset, especially when the models are trained using CD-1.

Due to space reasons, we show only the features of the SGRBM trained the MNIST dataset in Figure 4. Many features in Figure 4 look like different strokes of handwritten digits.

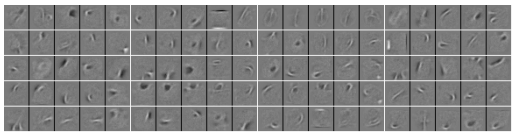
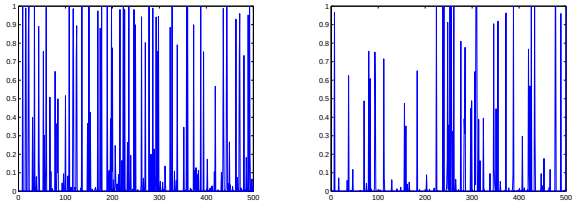


Figure 4: Learned features of the SGRBM trained on MNIST dataset. The white lines denote the boundaries of the groups.

We use Hoyer's sparseness measure (Hoyer 2004) to figure out how sparse representations learned by the RBMs and the SGRBMs. This sparseness measure is in the interval $[0, 1]$ and on a normalized scale. Its value more close to 1 means that there are more zero components in the vector. With every trained models, we can compute activation probabilities of hidden units given the test images. Given any trained model this leads to new representations of test data. Due to space reasons, we only report the results of the MNIST dataset[6]. The sparseness measures of the representations under the RBM (CD-1) trained on the MNIST dataset are in the interval $[0.52, 0.73]$, with an average of 0.64. The sparseness measures under the SGRBM (CD-1) are in [0.63, 0.80]. The averages are 0.72, respectively. It can be seen that the SGRBM can learn much more sparser representations. Figure 5(a) visualizes the activation probabilities of hidden units, which are computed under the regular RBM given an image from test set. Given the same image the activation probabilities computed under the SGRBM are shown in Figure 5(b).



(a) (b)

Figure 5: (a) Activation probabilities computed under the regular RBM. The sparseness of the vector is 0.64; (b) Activation probabilities computed under the SGRBM. The sparseness is 0.73.

Using SGRBMs to Pretrain Deep Networks

One of the most important applications of RBMs is to use them as building blocks to pretrain deep networks (Hinton and Salakhutdinov 2006). We show that SGRBMs can also be used to initialize deep networks and achieve better performances of classification on the MNIST dataset and the OCR letters dataset.

To make the comparison with the previous result (Hinton and Salakhutdinov 2006) fair, we use SGRBMs to pretain a 784-500-500-2000 network which has the same architecture to the deep network described in (Hinton and Salakhutdinov 2006). The group size and λ are selected in the way described in the previous section and set to 5 and 0.1. After pretraining, the multilayer network is fine-tuned using Conjugate Gradient. Then the network achieves the error rates of 0.96%. Using regular RBMs pretraining a 784-500-500-2000 network achieved the error rate of 1.14% (Hinton and Salakhutdinov 2006). A network with the same architecture initialized by sparse RBMs gave a much worse error rate of 1.87% (Swersky et al. 2010).

For the OCR letters dataset, we use SGRBMs to pretain a 128-1000-1000 network. All of hyperparameters are selected in the way described in the previous section. After fine-tuning, the network achieves the error rates of 9.79%. Using regular RBMs pretraining a network with the same architecture achieves the error rate of 11.21%.

Sparse Group Deep Boltzmann Machines

We also train a two layer (500-1000) sparse group Boltzmann machine on the MNIST dataset. We used the same hy-

[6]The results of the OCR letters dataset are similar

perparameters and the same training algorithm as Salakhutdinpov adopted in the source code[7]. The group size and λ are selected in the way described in the previous section and set to 10 and 0.1 for both of two layers. The SGDBM achieves the error rates of 0.84% on the test set, which is, to our knowledge, the best published result on the permutation-invariant version of the MNIST task. The deep Boltzmann machine with the same architecture resulted in the error rate of 0.95% (Salakhutdinov and Hinton 2009).

For the OCR letters dataset, we train a two layer (2000-2000) SGDBM. The group size and λ are set to 5 and 0.01 for both of two layers. We used the same hyperparameters as Larochelle adopted in the code[8]. The SGDBM achieves the error rates of 8.08% on the test set. The deep Boltzmann machine with the same architecture resulted in the error rate of 8.40% (Salakhutdinov and Larochelle 2010).

Conclusions

In this paper, we introduce l_1/l_2 regularizer on the activation probabilities of hidden units in RBMs, which lead to sparse group RBMs. Empirically we found SGRBMs can achieve better generative and discriminative performances than RBMs. At last, we illustrate the regularizer can also be applied to deep Boltzmann machines and improves discriminative performances of DBMs.

Acknowledgments This work was partially supported by National Natural Science Foundation of China (61071154, 60901078 and 60873132) and Zhengzhou Science & Technology (10LJRC189).

References

Bach, F. 2008. Consistency of the group Lasso and multiple kernel learning. *The Journal of Machine Learning Research* 9:1179–1225.

Bengio, Y. 2009. Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning* 2(1):1–127.

Freund, Y., and Haussler, D. 1994. Unsupervised learning of distributions on binary vectors using two layer networks. 912–919.

Garrigues, P., and Olshausen, B. 2008. Learning horizontal connections in a sparse coding model of natural images. In Platt, J.; Koller, D.; Singer, Y.; and Roweis, S., eds., *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press.

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

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

Hoyer, P. 2004. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research* 5:1457–1469.

Larochelle, H., and Bengio, Y. 2008. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, 536–543. ACM.

Lee, H.; Grosse, R.; Ranganath, R.; and Ng, A. 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. 2008. Sparse deep belief net model for visual area v2. In Platt, J.; Koller, D.; Singer, Y.; and Roweis, S., eds., *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press.

Nair, V., and Hinton, G. E. 2009. Implicit Mixtures of Restricted Boltzmann Machines. In Koller, D.; Schuurmans, D.; Bengio, Y.; and Bottou, L., eds., *Advances in Neural Information Processing Systems 21*. 1145–1152.

Olshausen, B., et al. 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381(6583):607–609.

Salakhutdinov, R., and Hinton, G. 2009. Deep Boltzmann machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 5, 448–455.

Salakhutdinov, R., and Larochelle, H. 2010. Efficient Learning of Deep Boltzmann Machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 6.

Salakhutdinov, R., and Murray, I. 2008. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th international conference on Machine learning*, 872–879. ACM.

Smolensky, P. 1986. Information processing in dynamical systems: foundations of harmony theory. In *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*, 194–281. MIT Press.

Swersky, K.; Chen, B.; Marlin, B.; and de Freitas, N. 2010. A tutorial on stochastic approximation algorithms for training Restricted Boltzmann Machines and Deep Belief Nets. In *Information Theory and Applications Workshop (ITA), 2010*, 1–10. IEEE.

Welling, M.; Rosen-Zvi, M.; and Hinton, G. 2005. Exponential family harmoniums with an application to information retrieval. In Saul, L. K.; Weiss, Y.; and Bottou, L., eds., *Advances in Neural Information Processing Systems 17*. Cambridge, MA: MIT Press. 1481–1488.

Yuan, M., and Lin, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(1):49–67.

[7] http://web.mit.edu/~rsalakhu/www/DBM.html

[8] http://www.cs.toronto.edu/~larocheh.