Boltzmann Machine and its Applications in Image Recognition

Shifei Ding, Jian Zhang, Nan Zhang, Yanlu Hou

Boltzmann Machine and its applications in image recognition

Shifei Ding[1, 2], Jian Zhang[1,2] Nan Zhang[1,2], Yanlu Hou[1,2]

1 School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, 221116, China
2 Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China
dingsf@cumt.edu.cn

Abstract. The overfitting problems commonly exist in neural networks and RBM models. In order to alleviate the overfitting problem, lots of research has been done. This paper built Weight uncertainty RBM model based on maximum likelihood estimation. And in the experimental section, this paper verified the effectiveness of the Weight uncertainty Deep Belief Network and the Weight uncertainty Deep Boltzmann Machine. In order to improve the images recognition ability, we introduce the spike-and-slab RBM (ssRBM) to our Weight uncertainty RBM and then build the Weight uncertainty spike-and-slab Deep Boltzmann Machine (wssDBM). The experiments showed that, the Weight uncertainty RBM, Weight uncertainty DBN and Weight uncertainty DBM were effective compared with the dropout method. At last, we validate the effectiveness of wssDBM in experimental section.

Keywords: RBM, DBM, DBN, Weight uncertainty

1 Introduction

The RBM is an unsupervised learning model which produces another expression of input data [1]. There are lots of training algorithms for RBM, such as Contrastive Divergence algorithm (CD), Persistent Markov chains and Mean Field methods, etc. In order to make full use of the features that extracted by RBM, Hinton et al built the DBN model [2-4]. The DBN model provides a feasible method to train Multilayer Perceptron by the process of unsupervised pre-training. Another classic model in deep learning field is the Deep Boltzmann Machine (DBM). DBM is powerful in image recognition and image reconstruction [5]. And there are many other powerful models in deep learning field [6-8]. The Extreme Learning Machine (ELM) and Multilayer Extreme Learning Machine performed well in classification problem [9]. In the field of image recognition, lots of research has been done as well [10-12].

Overfitting is a common problem in neural networks. To address this question, lots of algorithms are proposed. Dropout method is used to alleviate the overfitting problem, which can be used in training RBM as well [13]. However, according to our experiments, the Dropout RBM is not good at image reconstruction, although it is powerful in image recognition. The Weight uncertainty method is also widely used in neural networks to alleviate the overfitting problems [14]. In this paper, the weight random variables are used in training RBM to alleviate the overfitting problems. In our

experimental part, we validate the learning ability of Weight uncertainty RBM model. In classic RBM models, the conditional probabilities of visible units are binary. In Gaussian-binary RBM (mRBM) [15], the conditional probabilities of visible units follow Gaussian distribution. However, the mRBM performs not well in modeling nature images. In order to improve the images recognition ability, we introduce the spike-and-slab RBM (ssRBM) to our Weight uncertainty RBM and then build the Weight uncertainty spike-and-slab deep Boltzmann machine (wssDBM). At last, we validate the effectiveness of wssDBM in experimental section.

2 Restricted Boltzmann Machine and Semi-Restricted Boltzmann Machine

2.1. Restricted Boltzmann Machine models

RBM is a model based on energy functions. The structure of RBM is shown as Fig 1:

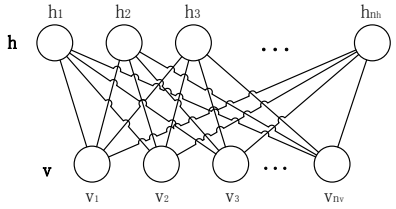

Fig 1. The structure of RBM

The RBM model consists of a visible layer and a hidden layer. If the visible units and the hidden units are binary, the energy function can be defined as follow:

$$\mathrm{E}(\vec{v},\vec{h}) = -\sum_{i=1}^{n_v} a_i v_i - \sum_{j=1}^{n_h} b_j h_j - \sum_{i=1}^{n_v}\sum_{j=1}^{n_h} h_j \times w_{ji} \times v_i \tag{1}$$

$\vec{a}$ is the bias vector of the visible layer, $\vec{b}$ is the bias vector of the hidden layer, W is the weight matrix between visible units and hidden units, $\vec{v}$ is the visible layer vector, $\vec{h}$ is the hidden layer vector. Then, the probability based on $E(\vec{v},\vec{h})$ is shown as formula 2:

$$P(\vec{v},\vec{h}) = \frac{1}{Z} e^{-E(\vec{v},\vec{h})} \tag{2}$$

Z is the partition function.

$$Z = \sum_{v,h} e^{-E(\vec{v},\vec{h})} \tag{3}$$

According to the whole training set, the likelihood function is defined as:

$$L_s = \ln \prod_{i=1}^{n_s} P(\vec{v^i}) = \sum_{i=1}^{n_s} \ln \mathrm{P}(\vec{v^i}) \tag{4}$$

n_s is the number of samples. And there are many algorithms can be used to maximize the likelihood function, such as Stochastic Gradient Descent algorithm. Let $\theta = \left(\vec{a},\vec{b},W\right)$, the derivative of the likelihood function is shown as formula 5:

$$\frac{\partial L_s}{\partial \theta} = -\sum_{i=1}^{n_s}\left(\sum_{h} P(\vec{h}\,|V^{(i)}) \frac{\partial E(V^{(i)},\vec{h})}{\partial \theta} + \sum_{v,h} P(\vec{v},\vec{h}) \frac{\partial E(\vec{v},\vec{h})}{\partial \theta}\right) \tag{5}$$

θ is the parameter. And the conditional probabilities are shown as follows:

$$p(h_k = 1 | \vec{v}) = sigmoid(b_k + \sum_{i=1}^{n_v} w_{ki} v_i) \tag{6}$$

$$p(\mathrm{v}_k = 1 | \vec{h}) = sigmoid(a_k + \sum_{j=1}^{n_v} h_j w_{kj}) \tag{7}$$

Hinton et al. proposed Contrastive Divergence (CD) algorithm to approximate the Maximum Likelihood Estimation. Based on single sample, and *k* is the number of steps in K-steps Contrastive Divergence algorithm (CD-K). We update the weights between visible units and hidden units with the following formulas:

$$\triangle w_{ij} = \eta_w (P(h_i = 1 | \vec{v}^{(0)}) \vec{v}^{(0)} - P(h_i = 1 | \vec{v}^{(k)}) \vec{v}^{(k)}) \tag{8}$$

$$\triangle a_i = \eta_a (v_i^{(0)} - v_i^{(k)}) \tag{9}$$

$$\Delta b_i = \frac{\partial \ln P(\vec{v})}{\partial b_i} \approx \eta_b \left(P(h_i = 1 | \overrightarrow{v^{(0)}}) - P(h_i = 1 | \overrightarrow{v^{(k)}}) \right) \tag{10}$$

η is the learning rate.

2.2. Spike-and-slab Restricted Boltzmann Machine

In order to model the expectation and covariance of Gaussian distribution, ssRBM model is proposed. In ssRBM, a variable slab is used to express the density. Based on the variable slab, the conditional probability of visible units has a diagonal covariance matrix. And the block Gibbs sampling can be used in ssRBM. The energy function can be expressed as follow:

$$E(v, s, h) = \frac{1}{2} v^T \Lambda v - \sum_{i=1}^{N} \left(v^T W_i s_i h_i + \frac{1}{2} s_i^T \alpha_i s_i + b_i h_i \right) \tag{11}$$

Beyond this energy function, the conditional probability can be expressed as follows:

$$p(v | h) = \frac{1}{B} N \left(0, \left(\Lambda - \sum_{i=1}^{N} h_i W_i \alpha_i^{-1} W_i^T \right)^{-1} \right) \tag{12}$$

$$p(s | v, h) = \prod_{i=1}^{N} N \left(h_i \alpha_i^{-1} W_i^T v, \alpha_i^{-1} \right) \tag{13}$$

$$P(h_i = 1 | v) = sigmoid \left(\frac{1}{2} v^T W_i \alpha_i^{-1} W_i^T v + b_i \right) \tag{14}$$

$$p(v | s, h) = N \left(\Lambda^{-1} \sum_{i=1}^{N} W_i s_i h_i, \Lambda^{-1} \right) \tag{15}$$

In this way, the conditional probability of visible units is a diagonal covariance matrix.

3 The training algorithms about RBM and Boltzmann Machine

There are lots of training algorithms for RBM model. Early, Persistent Markov chains and the Simulated Annealing method were used to estimate the data independent expectation and data dependent expectation. Although CD algorithm is not accurate in learning step-size, it guarantees the correct gradient direction. Based on CD algorithm, Persistent Contrastive Divergence algorithm (PCD) and Persistent Contrastive

Divergence algorithm with Fast weights (FPCD) are proposed. In order to decrease the sampling time in training process, Mean Field Method is proposed.

3.1. Mean Field Method

The detailed Mean Field Method is shown in reference. In the probabilistic graphical models, the real posterior distribution $P(h|v;\theta)$ is replaced by an approximate posterior distribution $Q(h|v;\lambda)$. What we need to do is minimizing the following *KL* Divergence:

$$\lambda^* = \arg \min_{\lambda} KL\left[Q(h|v) \| P(h|v)\right] \tag{16}$$

For the Mean Filed Boltzmann Machine, we have:

$$Q(h|v,u) = \prod_{i \in H} u^{S_i} (1-u_i)^{1-S_i} \tag{17}$$

Then, the KL Divergence can be expressed as follow:

$$KL[Q \| P] = \sum_i \left(u_i \ln u_i + (1-u_i)\ln(1-u_i)\right) - \sum_{i<j} \theta_{ij} u_i u_j - \sum_i \theta_i^c u_i + \ln Z_c \tag{18}$$

Z_c is the partition function, $Z_c = \sum_{\{H\}} \left(\exp\left(\sum_{i<j} \theta_{ij} S_i S_j + \sum_i \theta_i^c S_i \right) \right)$, $\theta_i^c = \theta_i + \sum_{j \in V} \theta_{ij} S_j$

S_i and S_j are independent variables. The expectation of S_i is u_i, and the expectation of S_j is u_j.

$$u_i = sigmoid\left(\sum_j \theta_{ij} u_j + \theta_i \right) \tag{19}$$

In general, the EM algorithm can be used in the Mean Field inference. Evidence shows that, for the same test data, the Mean Field Method is 10 to 30 times faster than Gibbs sampling. The Mean Field Method can be used in RBM model:

$$\ln P(v;\theta) \geq \sum_{i,j} W_{ij} v_i u_j + \sum_i b_i v_i - \ln Z - \sum_j \left(u_j \ln u_j + (1-u_j)\ln(1-u_j)\right) \tag{20}$$

The probability values of the hidden unit can be expressed as:

$$u_i = sigmoid\left(\sum_j W_{ij} v_j + b_i \right) \tag{21}$$

3.2. Persistent Markov chain

The detailed Persistent Markov chain algorithm is shown in reference [16, 17]. If the Markov chain is long, and the step-size is not too large, the Markov chain will reach the steady state. The Persistent Markov chains can be used in training Boltzmann Machines as well. For the data independence expectation, we can obtain an effective approximation. The algorithm of Persistent Markov chains is shown in Table 1:

Table 1. Algorithm of Persistent Markov chains

Algorithm of Persistent Markov chains
Randomly initialize θ_0 and M sample particles $\left\{\tilde{x}^{0,1}, ..., \tilde{x}^{0.M}\right\}$.
for $t = 0 : T$ (number of iterations) do
for $i = 1 : M$ (number of Parallel Markov chains) do
Sample $\tilde{x}^{t+1,i}$ given $\tilde{x}^{t,i}$ using transition operator $T_{\theta^t}\left(\tilde{x}^{t+1,i} \leftarrow \tilde{x}^{t,i}\right)$.

end for
Update: $\theta^{t+1} = \theta^t + \alpha_t \left[\Phi(\bar{x}) - \frac{1}{M} \sum_{m=1}^{M} \Phi\left(\tilde{x}^{t+1,m}\right) \right]$
Decrease α_t .
End for.

θ is a set of parameters, Φ is sufficient statistics vector, α_t is the learning rate.

4 Deep Belief Networks and Deep Boltzmann Machine

4.1 Deep Belief Networks

DBN is a hybrid network, which is proposed by Hinton in 2006. The top 2 layers consist of an associative memory with undirected connections. And the layers below have directed, top-down generative connections. In training process of DBN, the network is initialized layer by layer. Suppose that DBN is a model which has infinite layers. Then we use the same weight W_0 to initialize the network, the model can be considered as RBM in the training process, which is shown in Fig 2 (a). After training the first layer of DBN, the weights of the first layer remain constant, and the other weights are replaced by W_1. In this case, the priori information will be updated layer by layer. Hinton et al. proved that the pre-training process can tighten the variable boundary: $\ln p(v|W_1,W_2) \geq \ln p(v|W_1)$, and the pre-training process of DBN is shown as Fig 2 (b):

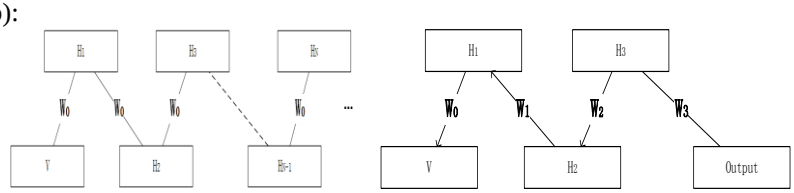

Fig 2(a) Fig 2(b)

Fig 2 shows the diagram of training process in DBN.

After pre-training in DBN, the DBN model can be fine-tuned by BP algorithm as a neural network.

4.2 Deep Boltzmann Machine

Different from DBN, DBM is still a Boltzmann Machine in topology. In the training process of DBM, the activation of each unit depends on the units in the adjacent layers. Salakhutdinov pointed out that the training process of DBM model can also be carried out layer by layer. However, different from DBN, Salakhutdinov showed that the different effects can be obtained by replacing the priori information with different proportions.

The probabilities in DBM model can be expressed as follows:

$$p\left(h_j^1 = 1 | \vec{v}, \overrightarrow{h^2}\right) = sigmoid\left(\sum_i W_{ij}^1 v_i + \sum_m W_{jm}^2 h_m^2 + b_j^1 \right) \tag{22}$$

$$p\left(h_m^2=1\mid\overrightarrow{h^1},\overrightarrow{h^3}\right)=sigmoid\left(\sum_j W_{jm}^2 h_j^1+\sum_l W_{ml}^3 h_l^3+b_m^2\right) \tag{23}$$

$$p\left(h_l^3=1\mid\overrightarrow{h^2}\right)=sigmoid\left(\sum_m W_{ml}^3 h_m^2+b_l^3\right) \tag{24}$$

$$p\left(v_i=1\mid\overrightarrow{h^1}\right)=sigmoid\left(\sum_j W_{ij}^1 h_j^1+b_j\right) \tag{25}$$

The superscripts represent the layer number. The log-likelihood can be approximated by using Stochastic Approximation algorithm and Mean Field Algorithm.

5 Weight uncertainty spike-and-slab Restricted Boltzmann Machine

5.1. Weight uncertainty method

In the whole training process, the weights and biases need to be calculated. And the weights and biases are regarded as real valued variables. In this case, training a neural network prefer to encounter the problem of overfitting. There are lots of research about alleviating the overfitting problem in neural networks. Based on RBMs, the main algorithm is dropout method. Although dropout RBM is useful to alleviate the overfitting problem in classification, the image reconstruction ability of dropout RBM is not better than conventional RBM. If the weights are considered as random variables, the above problems may be alleviated. The weights are considered as random variables, and we assume that the random variables follow Gaussian distribution. What we need to do is calculating the expectation and the covariance. And the generations of different weights can be regarded as the sampling from Gaussian distribution. Therefore, the Weight uncertainty neural network can be considered as the ensemble of neural networks.

In the research of Blundell et.al, all weights in networks are regarded as probability distributions, rather than a real value. The objective is to find a variational approximation to the Bayesian posterior distribution on the weights. And the objective function can be expressed as follows:

$$\begin{aligned}\theta' &= \arg\min_\theta KL\left[q(w\mid\theta)\,\|\,P(w\mid\theta)\right]\\ &= \arg\min_\theta KL\left[q(w\mid\theta)\,\|\,P(w)\right]-E_{q(w\mid\theta)}\left[\log P(D\mid w)\right]\end{aligned} \tag{26}$$

According to the thought of MAP estimation, let

$$f(w,\theta)=\log q(w\mid\theta)-\log P(w)P(D\mid w) \tag{27}$$

In RBM model, in order to obtain more effective image reconstruction and classification, we introduce the weight random variables to RBM model. The cost functions of RBM can be written as: maximum likelihood estimation $p(v\mid w)$, and MAP estimation $p(w\mid v)$. In order to simplify calculation, we use the Maximum Likelihood Estimation to calculate the activation probabilities. Assuming the weight W follows the Gaussian distribution, the mean value of W is μ, standard deviations are $\sigma=\log(1+\exp(\rho))$, if $\varepsilon\sim N(0,I)$, the weights can be expressed as: $w=\mu+\log(1+\exp(\rho))\circ\varepsilon$. According to the chain rule, the derivatives can be expressed as follows:

$$\frac{\partial \log p(w_{ij})}{\partial w_{ij}} \times \frac{\partial w_{ij}}{\partial \mu} = \left(P\left(h_j = 1 | \vec{v}\right) v_i - \sum_{v,h} P(v) P\left(h_j = 1 | \vec{v}\right) v_i \right) \times 1 \tag{28}$$

$$\frac{\partial \log p(w)}{\partial w} \times \frac{\partial w}{\partial \rho} = \left(P\left(h = 1 | \vec{v}\right) v - \sum P(v) P\left(h = 1 | \vec{v}\right) v \right) \times \frac{\varepsilon}{1 + \exp(-\rho)} \tag{29}$$

In the experimental section, we test the classification ability and image reconstruction ability of Weight uncertainty RBM model (WRBM), and then build the DBN and DBM based on WRBM.

5.2. Weight uncertainty spike-and-slab deep Boltzmann Machine

ssRBM is used to model nature images. In this paper, we use the ssRBM as the feature extractor, and build the DBM model, and then we introduce the weight random variables to the DBM, and build the wssDBM. At last, we validate the effectiveness of wssDBM in experimental section.

6 Experimental analysis

Firstly we compare the WRBM with RBM and dropout RBM in classification and image reconstruction. The algorithm we used in fine-tuning process is the conjugate gradient algorithm, the iterative steps are 100. In this experiment, we use MNIST, MNIST-Basic and Rectangles as the testing data sets. The attributes of these data sets are shown in Table 2:

Table 2. The attributes of data sets

	Number of training samples	**Number of testing samples**	**attributes**	**labels**
MNIST-Basic	10000	50000	784	10
Rectangles	1000	50000	784	2
MNIST	60000	10000	784	10

Firstly we test the image recognition ability of the WRBM (WRBM). In fine-tuning process. The testing accuracies are shown in Table 3:

Table 3. The number of misclassifications in shallow models

	MNIST-Basic	**Rectangles**
RBM-BP	1811	2586
Dropout RBM-BP	1633	2175
WRBM-BP	1567	1979

As we can see from Table 3, the classification accuracies of WRBM are better than RBM and dropout RBM, that is to say, like dropout method, the weight random variables are useful in classification problems.

The reconstruction errors in training process are shown in Table 4:

Table 4. The reconstruction errors of RBM and WRBM

	MNIST-Basic
RBM	61631
Dropout RBM	65623

WRBM	52638

As we can see from Table 4, the image reconstruction ability of WRBM is better than other models. And the weight random variables are also useful in image reconstruction.

The topologies in DBM and Weight uncertainty DBM (WDBM) are 784-1000-1000-10. And the topologies in DBN and Weight uncertainty DBN (WDBN) are 784-1000-2000-10. The iterative steps in RBM training process are 200. The iterative steps in DBM training process is 300. The testing accuracies are shown in Table 5.

Table 5. The number of misclassifications of DBN and DBM

	MNIST-Basic	**MNIST**	**Rectangles**
DBM	1115	94	1309
WDBM	1016	92	1139
DBN	1283	105	1278
WDBN	1251	101	367
Dropout DBN	1257	99	778
wssDBM	1022	103	477

As we can see from Table 5, the WDBM performs better than conventional DBM model, and WDBN is also comparable to dropout DBN in classification problem. wssDBM also performs well in classification problems.

7 Conclusion

In this paper, in order to alleviate the overfitting problem, and improve the ability of image reconstruction in RBM model, we introduce the Weight uncertainty method to RBM. The WRBM performs well in our experiments. In our experiments, the Weight uncertainty method is useful in both classification and image reconstruction. Intuitively speaking, the weight random variables can be regarded as the ensemble of neural networks. And the wssDBM is useful in image recognition.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61379101, 61672522), and the National Key Basic Research Program of China (No. 2013CB329502).

References

1. G.E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence, Neural Computation, 2002, 14(8): 1771-1800.
2. N. Roux, Y. Bengio. Representational Power of Restricted Boltzmann Machines and Deep Belief Networks, Neural Computation, 2008, 20(6): 1631-1649.
3. G.E. Hinton, S. Osindero, Y. Teh. A fast learning algorithm for deep belief nets, Neural Computation, 2006, 18(7): 1527-1554.
4. G. E. Hinton, R. Salakhutdinov. Reducing the dimensionality of data with neural networks, Science, 2006, 313(5786):504-507.

5. H. Lee, P. T. Pham, L. Yan, et al. Unsupervised feature learning for audio classification using convolutional deep belief networks, Advances in Neural Information Processing Systems, 2009:1096-1104.
6. M. Norouzi, M. Ranjbar, G. Mori. Stacks of convolutional Restricted Boltzmann Machines for shift-invariant feature learning, Computer Vision and Pattern Recognition, 2009:2735-2742.
7. R. Salakhutdinov, H. Larochelle. Efficient Learning of Deep Boltzmann Machines. Efficient Learning of Deep Boltzmann Machines, Journal of Machine Learning Research, 2010, 9(8):693-700.
8. R. Salakhutdinov, G.E. Hinton. An Efficient Learning Procedure for Deep Boltzmann Machines, Neural Computation, 2012, 24(8):1967 - 2006.
9. J. Zhang, S. F. Ding, N. Zhang, et al. Incremental extreme learning machine based on deep feature embedded, International Journal of Machine Learning and Cybernetics, 2015, to be publised.
10. N. Zhang, S. F. Ding, Z. Z. Shi. Denoising Laplacian multi-layer extreme learning machine, Neurocomputing, 2016, 171: 1066–1074, to be published.
11. S. F. Ding, N. Zhang, X. Z. Xu, et al. Deep Extreme Learning Machine and Its Application in EEG Classification, Mathematical Problems in Engineering, 2015, 1-11.
12. Y. Zheng, B. Jeon, D. Xu, Q.M, et al. "Image segmentation by generalized hierarchical fuzzy C-means algorithm," Journal of Intelligent and Fuzzy Systems, 2015, 28(2): 961-973.
13. N. Srivastava, G.E. Hinton, A. Krizhevsky. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, 2014, 15:1929-1958.
14. C. Blundell, J. Cornebise, K. Kavukcuoglu. Weight uncertainty in Neural Networks, Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015.
15. A. Krizhevsky, G.E. Hinton. Learning multiple layers of features from tiny images, Technical report, U. Toronto, 2009.
16. T. Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient, Proceedings of the 25th International Conference on Machine Learning. ACM, 2008: 1064-1071.
17. T. Tieleman, G.E. Hinton. Using fast weights to improve persistent contrastive divergence, Proceedings of the 26th International Conference on Machine Learning. ACM, 2009: 1033-1040.