Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **6**, November **2016**

The Classification of Synthetic Aperture Radar Image Target Based on Deep Learning

Xiu-Yuan Chen, Xi-Yuan Peng, Yu Peng and Jun-Bao Li

Department of Automatic Test and Control
Harbin Institute of Technology
92 Xidazhi Street, Harbin, 150080, China
hitchenxiuyuan@163.com; pxy@hit.edu.cn; pony911@163.com; lijunbao@hit.edu.cn

Received May, 2016; revised June, 2016

Abstract. *In recent years, Deep learning is used to achieve its efficient performance in tasks such as object perception and image target recognition. Multilayer neural network model is one of the most basic and wide-used learning models in deep learning algorithm. The SAR (Synthetic Aperture Radar) image is widely used in national economy and military areas. This paper introduces Deep Belief Network to realize the classification of SAR image target by training every layer respectively with Restricted Boltzmann Machine and using the back propagation algorithm to adjust parameters of the whole network.*
Keywords: Deep Belief Network, Restricted Boltzmann Machine, SAR Image, Target Classification

1. **Introduction.** Image recognition and classification has been a popular area of research for several years, and according to different requirements, it has been used in many aspects, such as military, healthcare, and transportation field. The critical factor for human beings to survive in the natural world is the human visual system, which helps people recognize and classify targets. In order to mimic this process and have an outstanding performance in automatic image recognition and classification, scientists proposed deep learning and have proved its extensive application in different areas.

Deep learning is a machine learning method based on learning representations of data which makes it easy to learn tasks[1]. Deep learning is a method of multilayer structure, which extracts the data feature from low-layer to high-layer gradually, and eventually forms the ideal characteristics for identify to improve the accuracy of classification or recognition. The earliest deep learning method called DBN (Deep Belief Network) was proposed by Hinton in 2006[2], in order to solve the problem of image classification. Their experiments were based on the MNIST dataset, which contains 60,000 training images and 10,000 test images. On these images, the result proved that the DBN performed well theoretically, and it can be used in target classification for general images. So we introduce the DBN in our paper to realize the classification of SAR (Synthetic Aperture Radar) images and verify how well the DBN can do in this task.

The SAR provides high-resolution radar image and its widely used since the SAR works well even under the condition of complex environment[3]. Though the SAR image has advantages in several kinds of radar image, SAR image classification is a high-dimensional nonlinear mapping problem. Beyond that the image noise and the high-dimensional character make it a big challenge to manually extract features, so we introduce deep learning to automatically extract features from image data. The traditional SAR classification methods are based on statistical distribution. Bayes classifier based on complex Gaussian distribution for SAR classification[4]. Then with the development of machine learning, SVM was introduced to solve SAR object classification[5]. However, most of the researches aimed at heterogeneous land cover[6][7].

This paper takes the tank images from the MSTAR image set as the research object of SAR image target classification and there are 1,800 training images and 400 test images in our dataset, and the number is big enough for the DBN to learn and classify. The rest of this paper will contain the following content. Section 2 introduces the architecture of the Deep Belief Network and the process of training. In Section 3, we show how we structure our dataset from the MSTAR image set and we add salt-and-pepper noise and Gaussian noise to verify the influence of the differences in the quality of the SAR images. Our experiment and operation of training and testing image target will be shown in section 4. We mainly discuss the influence of layer number and unit number for accuracy. And in section 5 is the conclusion and discussion from the testing result.

2. **Deep Belief Network.** The Deep Belief Network is composed of several hidden layers of hidden units with connections between the layers but not between units within each layer. When training the images in an unsupervised way, the DBN learns to reconstruct the image as input vectors, and the hidden layers act as feature detectors on inputs, so that the DBN can be used to perform classification[8]. Figure 1 shows the construction of the Deep Belief Network in our experiment. In our experiment, each training case consists of an image and a certain label. The size of images and the number of labels can be changed with different target. We train the network layer-by-layer and use the RBM (Restricted Boltzmann Machines) to train every layer[9]. The parameters of the whole network are adjusted together after the layer-by-layer training.

The dashed boxes in Figure 1 are the basic learning units called RBM. An RBM is a graphical model and all its nodes are divided into two sets: visible units (node) and hidden units (node). Each visible unit is connected to each hidden unit, however, there are no connections between the visible units and the hidden units are also disconnected from each other[10]. Figure 2 shows the construction of a basic RBM. The stochastic variables from 0 to 1 in the hidden layer are connected to the stochastic variables in the visible layer and there are no connections inside a layer. When several RBMs compose the deep belief network, the higher-level RBM is trained by using the hidden activities of the lower RBM as data.

In image target classification, the pixels of image correspond to the visible units and feature detectors correspond to the hidden units. The energy of an RBM with V visible units and H hidden units is given by the following function:

$$E(v,h) = -\sum_{i=1}^{V}\sum_{j=1}^{H} v_i h_j w_{ij} - \sum_{i=1}^{V} v_i b_i - \sum_{j=1}^{H} h_j b_j \tag{1}$$

Where v and h are the binary state vectors of the visible and hidden units, vi and hj are the states of visible and hidden units, and wij is the real-valued weight between them. bi and bj are the bias of visible and hidden units. The neural network gives all the probable images a probability with the function above, and the probability of the image can be raised by lower the energy of it. Given a training image, the probability of the binary state hj of each hidden unit is set to 1 by the function:

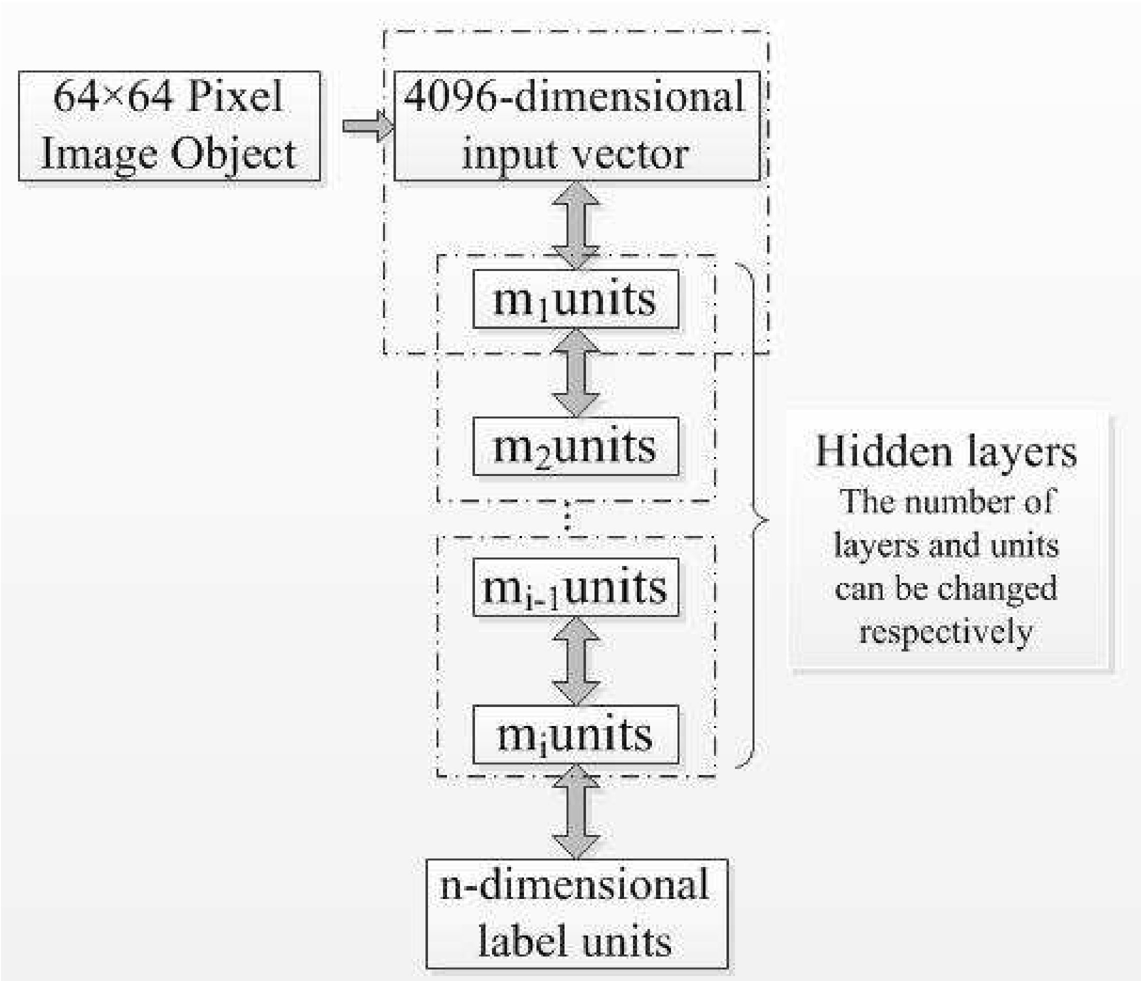

FIGURE 1. The construction of Deep Belief Network

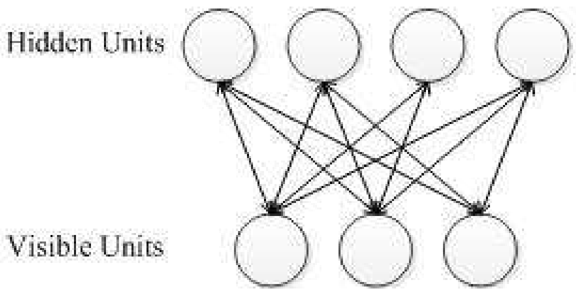

FIGURE 2. The basic structure of RBM

$$p(h_j = 1|v) = \sigma(b_j + \sum_{i=1}^{n} v_i w_{ij}) \quad (2)$$

Where (x) is the logistic function1/[1+exp(-x)], bjis the bias of unit j, vi is the state of unit i, and wij is the weight between i and j. When the states of the hidden units have been chosen, the probability of setting each vi to 1 is decided by the function:

$$p(v_i = 1|h) = \sigma(a_i + \sum_{i=1}^{n} w_{ij}h_j) \quad (3)$$

Since the units in the visible layer are independent from each other and it is the same between the hidden units, the p of forward propagation and reconstruction are given by the function:

$$p(h|v) = \prod_{j}^{m} p(h_j|v) \quad (4)$$

$$p(v|h) = \prod_{i}^{n} p(v_i|h) \quad (5)$$

During the learning processing of the RBM, weight and bias are updated in each iteration epoch at the same time, and their update is given by the following functions:

$$W_t = mW_{t-1} + \varepsilon(v \cdot p(h|v) - v' \cdot p(h'|v')) \quad (6)$$

$$b_t = mb_{t-1} + \varepsilon(p(h|v) - p(h'|v')) \quad (7)$$

$$a_t = ma_{t-1} + \varepsilon(p(v|h) - p(v'|h')) \quad (8)$$

Where m is the momentum which is set to 0.5 initially. When the reconstruction error nearly changes to be steady, the momentum is set to 0.9 and is a learning rate during the update. Learning layer-by-layer is efficient but not optimal enough. Once the weights of higher layers have been learned, the weights may not be optimal for the lower layers. So we use the back propagation to adjust the weight of the deep belief network[11]. The BP algorithm is a supervised classifier which can classify the feature vectors from the pre-training by RBMs and do the fine-tuning of DBNs variations at the same time. Compare the practical classification and the labels to get the error,and the error is back propagated to adjusted the parameters of the DBN. During the backpropagation, we calculateof each layer. For the output layer, the practical output is oi, and the expected output is di, then the parameter is decided by the function:

$$\delta_i = o_i(1 - o_i)(d_i - o_i) \quad (9)$$

Then during the backpropagation, we can get the of hidden layers from the following function:

$$\delta_i^l = y_i^l(1 - y_i^l) \sum_j w_{ij}^l \delta_j^{l+1} \quad (10)$$

After the backpropagation and the of all layers are collected, they are used to correct the weight parameters of the network, and the weights are updated by the following functions:

$$w_{ij}^l = w_{ij}^l + \varepsilon_{fine-tuning} \times y_i^l \delta_j^{l+1} \quad (11)$$

$$b_j^l = b_j^l + \varepsilon_{fine-tuning} \times \delta_j^{l+1} \quad (12)$$

3. **Dataset.** In order to test the classification of SAR images based on the Deep Belief Network, we picked three kinds of tank images from the MSTAR image set to form the dataset used in our experiments. Our dataset includes 2200 images and they are all taken from the 0o angel and 15o angel. Some of the SAR images we use in our target classification are shown in Figure 3. These SAR images have different sizes and different amount of pixels. Since the input matrix consists of vectors composed of all the pixels in test images and the DBN in our experiment requires that the vectors have same dimension. The images in the dataset should be used after pre-processing. In addition, more pixels mean more elements in the vectors, and that may increase the operation time. So we pre-process the images and keep 64*64 pixels in every image. Figure 4 shows some of the images with 64*64 pixels from the dataset we used in our experiment. We do this pre-processing to reduce the pixels in each image so that the dimension of the input becomes smaller, which makes the operation time shorter. During the research, 1800 of them are used as training images, and 400 images are for the test.

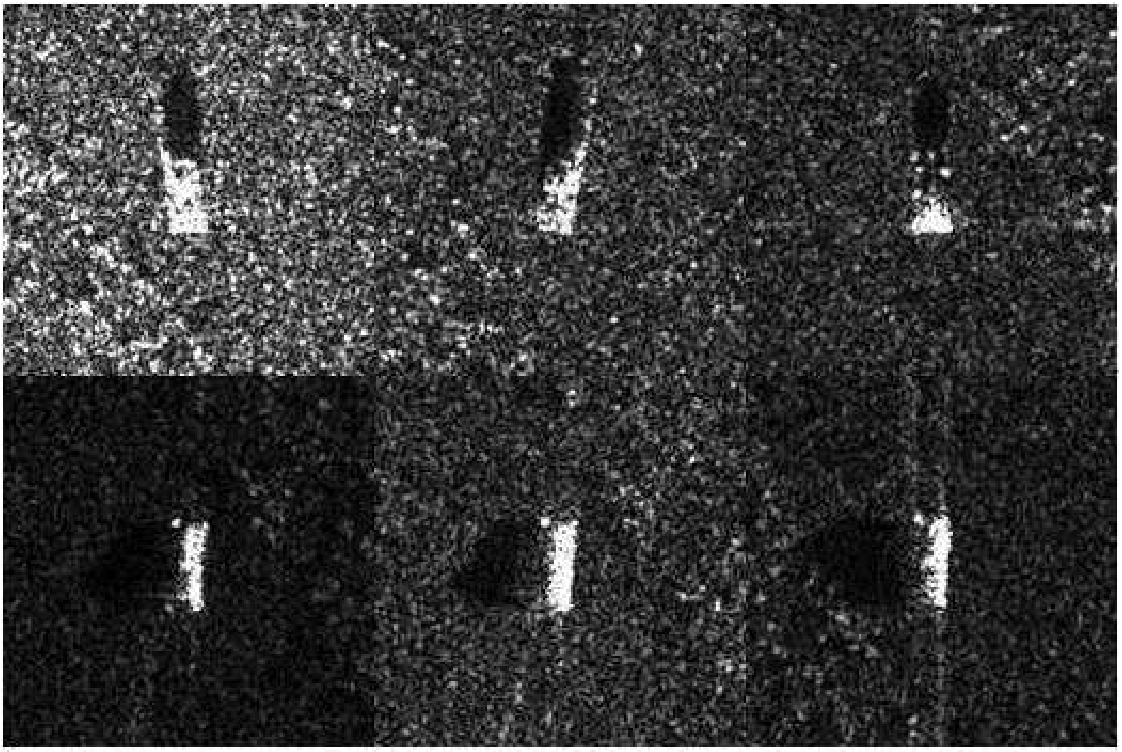

FIGURE 3. The SAR images from MSTAR database

4. **Experiment and operation.** The idea of Deep Belief Network is to train the network layer by layer and then use the backpropagation to do the fine-tuning. Therefore, for a l-layers DBN, we considered the following training procedure:

(1)Read the original images and process them into vectors. The input of network is the matrix composed of these vectors.

(2)Load the input data and prepare their labels for the backpropagation.

(3)Repeat l times:

(a)Confirm the number of input and output units of the layer.

(b)train with the RBM and save weight and bias of the layer.

(4)Do the backpropagation and output the error rate of the classification.

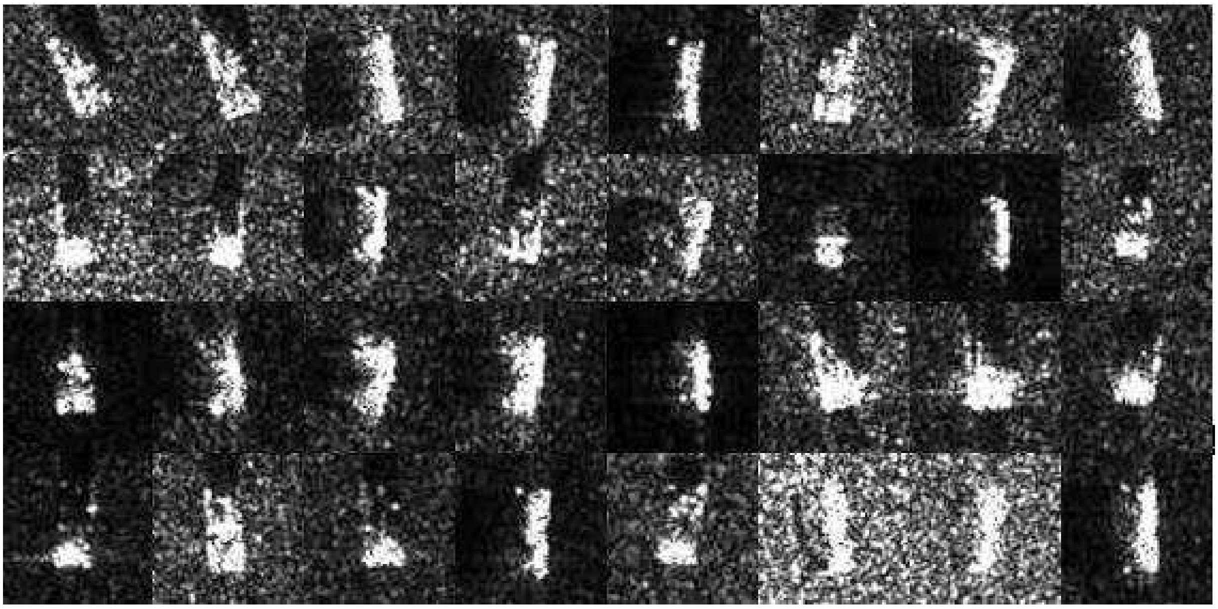

FIGURE 4. Some of the images used after pre-processing

The performance of Deep Belief Network is influenced by many factors such as the number of layers, the number of units in every layer, the learning rate and so on. We use steepest descent and a small learning rate, which are fixed in our experiments. In this paper, our discussion focuses on the number of layers and the units in every layer. The classification result of four different DBNs is shown in table 1. Especially, we set 500 units in the layer connected to input, and 2000 units in the layer connected to output. The hidden layers inside have 1000 units respectively.

TABLE 1. The result of DBNs with different numbers of layers

Num of layers	Units of layers	Time(h)	Error rate
2	500-2000	0.685	14%
3	500-1000-2000	1.069	6.5%
4	500-1000-1000-2000	1.011	7.5%
5	500-1000-1000-1000-2000	1.358	12.8%

The data in the third line shows that with the increase of the layer number, the operation time increases. Since the multilayer structure extracts the data feature from low-layer to high-layer gradually and obtains the more abstract feature of objects, researchers anticipate that more layers in the network means better performance in tasks. However, our research proves that a DBN with 3 or 4 hidden layers works better than the others, which has also been proved by other researchers when they used DBN doing their work. And in our classification of SAR images with the 3 layers DBN, the error rate reaches 6.5%, while 4 layers DBN has a similar accuracy.

Table 2 shows the result of DBNs with different numbers of units. We change the number in three strategies: 1)we change the number of units of middle hidden layers. 2)we change the number of units of all the hidden layers. 3) we change the number of units of the hidden layer connected to output layer. Experiment a is the experiment we do above which aims at a 4-layer DBN. From table 2, we can see that as the number of units decreases, the operation time gets short correspondingly. However, the change of error

Table 2. The result of DBNs with different numbers of units

Experiment	Units of layers	Time(h)	Error rate
a	500-1000-1000-2000	1.011	7.5%
b	500-500-500-2000	0.505	11.8%
c	1000-2000-2000-4000	2.723	15%
d	500-500-500-500	0.391	15.8%

rate brought by the change of unit number is more complicated. Compare experiment a with b, and there difference shows that less hidden layer units can reduce the operation time, while the error rate raises obviously. Generally more units in the hidden layers can make the network extract more features of objects. Yet the comparison of experiment a and c shows that excess units fail to promote the performance of network, while they extend the operation time and raise the error rate on the contrary. Since theres no standard to follow to set the number of units in different applications, we have to try different combinations to find the optimum choice.

Besides the experiments discussed above, we have also do the classification on SVM to compare the effect of deep network and shallow network. Since the SVM initially came into use, researchers have introduced kernel functions to improve its non-linear effect. The SVM performs differently with different kernel function, so we select three kernel functions and compare them with Deep Belief Network.

Table 3. The result of SVM and the comparison to DBN

Experiment	Classification Method	Error rate
a	4-layer DBN	7.5%
b	Polynomial-SVM	55.5%
c	RBF-SVM	16.5%
d	Sigmoid-SVM	26%

TABLE 3 shows the classification result of SVM and the comparison between DBN and SVM. In this part, we use SVM with three kinds of kernel functions to do the same classification of SAR images: 1)SVM with Polynomial kernel function. 2) SVM with RBF kernel function. 3) SVM with Sigmoid kernel function. Experiment a is the experiment we do above which aims at a 4-layer DBN. SVM is the most common shallow network method in machine learning, and it uses different kernel functions to promote its non-linear behavior in applications. However, when these kernel functions are applied to the classification of SAR images, we can see from the result that their non-linear feature extraction ability is unsatisfied. The highest error rate reaches 55.5% when we use the polynomial-SVM, and the best result comes from RBF-SVM, but its error rate is still more than twice the result of 4-layer DBN.

5. **Conclusion.** Deep network models are efficient in representing complex functions but very difficult to train. The Deep Belief Network has solved this problem by layer-by-layer pre-training and fine-tuning the network integrally with backpropagation. According to the result of the experiments we do in our research on SAR image classification, the error rate of Deep Belief Network with the structure of 4096-500-1000-2000 and 4096-500-1000-1000-1000-2000 are lower than the other situations. And the classification effect of deep network is better than the shallow network(such as the three kinds of SVMs used).

TABLE 4. The classification result of all the constructions in our experiments

No.	Classification Method	Constr.	Units parameters	Error rate
1	2-layer DBN	Deep	500-500	14.8%
2	2-layer DBN	Deep	500-1000	12.3%
3	2-layer DBN	Deep	500-2000	14%
4	3-layer DBN	Deep	500-500-500	16%
5	3-layer DBN	Deep	500-500-2000	6.7%
6	3-layer DBN	Deep	500-1000-2000	6.5%
7	3-layer DBN	Deep	1000-2000-4000	11.3%
8	4-layer DBN	Deep	500-500-500-500	15.8%
9	4-layer DBN	Deep	500-500-500-2000	11.8%
10	4-layer DBN	Deep	500-1000-1000-2000	7.5%
11	4-layer DBN	Deep	1000-2000-2000-4000	15%
12	5-layer DBN	Deep	500-500-500-500-500	15.2%
13	5-layer DBN	Deep	500-500-500-500-2000	10.5%
14	5-layer DBN	Deep	500-1000-1000-1000-2000	12.8%
15	Polynomial-SVM	Shallow	———	55.5%
16	RBF-SVM	Shallow	———	16.5%
17	Sigmoid-SVM	Shallow	———	26%

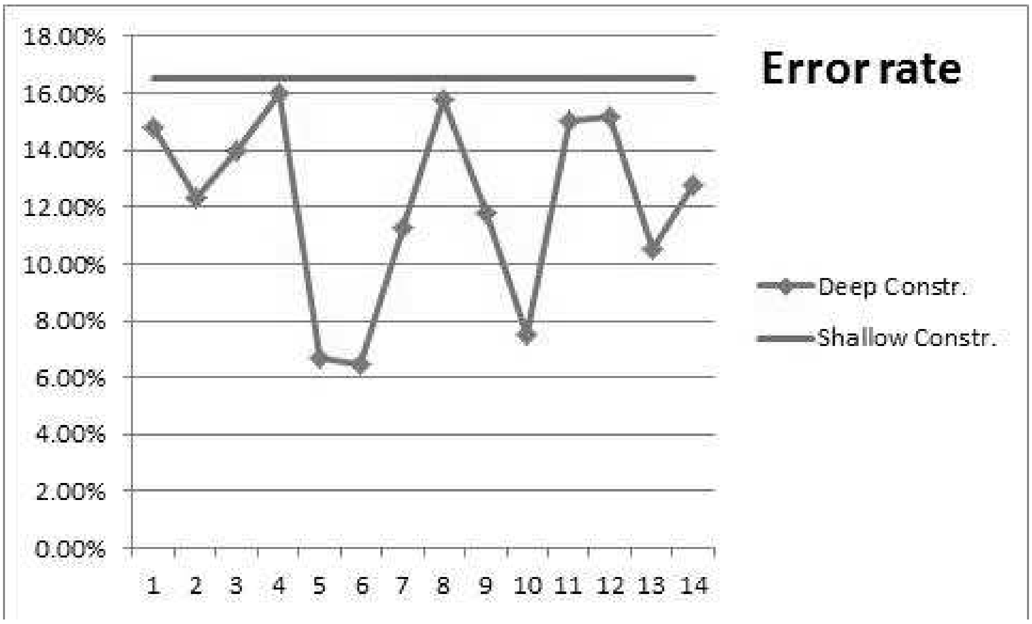

FIGURE 5. The comparison of error rate between deep network and the best result of shallow construction

Nonetheless, the parameters of DBN are adjusted without clear standards, which means we cannot exactly know in what situation we can get the best classification results. Besides, more deep network structures have been proposed in recent years, such as Convolutional Neural Network (CNN). We will try other deep network structures and find a more efficient way to perform the classification of SAR images and lower the error rate as far as possible.

Acknowledgment. This work is supported by Program for New Century Excellent Talents in University under Grant No. NCET-13-0168, National Science Foundation of China under Grant No.61371178, Heilongjiang Science Foundation of China under Grant No.F2015003, Harbin Science Foundation of China under Grant No. 2015RQQXJ088. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] J. Schmidhuber, Deep Learning in Neural Networks: An Overview, *Neural Networks*, vol. 61, pp. 85–117, 2015.

[2] G. Hinton, S. Osindero, Y, and Teh, A Fast Learning Algorithm for Deep Belief Nets, *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[3] K. Arai, Y. Terayama, and T. Arata, Image classification based on beta distribution for SAR image, *Geoscience and Remote Sensing Symposium, 1995. IGARSS '95. 'Quantitative Remote Sensing for Science and Applications', International. IEEE*, vol. 2, pp. 1263–1265, 1995.

[4] J. A. Kong, A. A. Swartz, H. A. Yueh, and L. M. Novak, Identification of terrain cover using the optimum polarimetric classifier, *Journal of Electromagnetic Waves and Applications*, Kaohsiung, vol. 2, no. 2, pp. 171–194, 1988.

[5] C. Yuan, D. Casasent, A new SVM for distorted SAR object classification, *Proceedings of SPIE - The International Society for Optical Engineering*, 2005.

[6] G. Vasile, J. Ovarlez, F. Pascal, and C. Tison, Coherency matrix estimation of heterogeneous clutterin high-resolution polarimetric SAR images, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, pp. 1809–1826, 2010.

[7] L. Bombrun, G. Vasile, M. Gay, and F. Totir, Hierarchical segmentation of polarimetric SAR images using heterogeneous clutter models, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, pp. 726–737, 2011.

[8] G. Hinton, Deep belief networks, *Scholarpedia*, vol. 4, no. 6, pp. 786-804, 2009.

[9] G.E. Hinton, and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, vol. 313, pp. 504-507, 2006.

[10] G.E. Hinton, Learning multiple layers of representation, *Trends in Cognitive Sciences*, vol.11, no.10, pp.428-434, 2007.

[11] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by backpropagating errors, *Nature*, vol. 323, pp.533-536, 1986.