A new node selection method of Deep Belief Network based on similarity

Leifang Yan[1, a], Mingyan Jiang[2,b*]

[1,2]School of Information Science and Engineering,Shandong University, Jinan 250100, China

[a] mochaoxygen2@126.com, [b*]corresponding author: jiangmingyan@sdu.edu.cn

Keywords: deep belief network; node selection; similarity; deep learning; unsupervised learning

Abstract. Research on the node selection problem in Deep Belief Network(DBN). A new node selection model based on similarity is proposed to select nodes in the pre-training process of DBN, which is an unsupervised process. Firstly, extracting the nodes feature from pre-training weight matrix, and then comparing them each other to get the similarity matrix, and using the similarity matrix to select nodes and reusing the nodes. Finally,the test on MINIST data set proves that the new model has better performance.

Introduction

Since 2006, deep structured learning, or more commonly called deep learning or hierarchical learning, has emerged as a new area of machine learning research[1] . DBN is one of the most popular topics in deep learning. DBN proposed by Hinton et al. is a new learning algorithm of multi-layer neural network[2]. DBN is stacked by layers of Restricted Boltzmann Machine(RBM), and the training algorithm of DBN is the same with RBM, called Constrastive Divergence(CD). Recently, research on DBN mostly focuses on optimizing CD algorithm. Persistent Contrastive Divergence algorithm proposed by Tijmen Tieleman solves the problem of maximizing the likelihood of CD[3] . And then a series of algorithms based on Markov Chain Monte Carlo(MCMC) are proposed to optimize DBN algorithm[4] . In the training process ,we set up the network selecting the depth and the number of neurons according to experience before[5] .

In this paper, we propose a new method to solve the neurons problem for the first time. Our work focuses on reusing neurons to get full use of all the neurons after applying CD to train the network.

Deep Belief Network

Pre-training. DBN is constructed by layers of RBMs. The training process of DBN is from bottom to top. 1) the input of bottom RBM is original data, 2) the input of the top RBM is the feature exacted by the bottom RBM[6]. A RBM model is a particular type of Markov random field that has a two-layer architecture, in which one is visible and the other is hidden[7] . As Fig.1 shows, there are links between the nodes in different layers, while there are no links between nodes in a single layer.

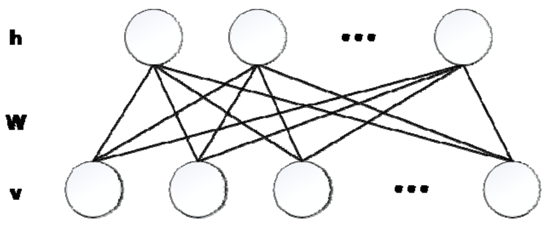

Fig. 1. A typical RBM with two layers.

RBM is defined by energy, we discuss its energy function and the related probability distribution.

$$E(v,h) = \sum_{i\in v} a_i v_i \sum_{j\in h} b_j h_j \sum_{i\in v, j\in h} w_{ij} v_i h_j. \tag{1}$$

Above, a_i is the biases of visible layer and b_j is the biases of hidden layer,and w_{ij} is the weights between the node i in visible and the node j in hidden layer. And given the energy function, we can get the joint probability in the state (v,h) .

$$P(v,h)=\frac{1}{Z}\exp(\quad E(v,h))$$
$$Z=\sum_{v,h}\exp(\quad E(v,h)). \tag{2}$$

In the training process, we focus on the marginal distribution of visible layer, it stands for the imitative effect to the input data, the bigger the better.

$$P(v)=\sum_{h}P(v,h)=\frac{1}{Z}\sum_{h}\exp(\quad E(v,h)). \tag{3}$$

In order to maximize the visible layer's marginal distribution P(v), we consider changing the parameters in the equation. P(v) depends on E(v,h), and E(v,h) depends on the parameters of two layers, namely a_i , b_j , w_{ij} . So what we should do is to adjust these parameters of the network to maximize the marginal distribution. Due to the complex partition function E, it is hard to calculate. And then CD algorithm is proposed to solve the problem.CD algorithm takes advantage of a group of the sample data as the training data, and through approximating simplifies the problem.

Fine-tuning. Through the pre-training process, we get the parameters of the stacked RBMs, and we assignment all the parameters to a Back Propagation Neural Network(BP). And BP adjust the parameters of the whole structure to get the optimal output.

New methods based on similarity

The method we determine the number of hidden layer nodes is by experience , leading to more nodes than we need and increasing the complexity of networks. To make full advantage of all the nodes, we propose a new method to select nodes applying to DBN.

The new method inspires by the sparse coding algorithm. Sparse coding is a learning algorithm mainly for unsupervised feature for finding succinct, a little above high- level representation of inputs, and it has successfully given a way for Deep learning[8]. We select a few base vectors from the base dictionary to encode the original data, and only the selected base vectors have strong response.

$$I(x,y)=\sum_{i}a_i\varphi_i(x,y). \tag{4}$$

Above , I(x, y) is the input data, φ(x,y) is the base vector and a_i is the coding coefficient. The number of hidden layer nodes is i, the node set is A. And we select a representative set A_r and k, l < i.

$$A=\{A_1,A_2,A_3,\dots,A_i\}. \tag{5}$$
$$A_r=\{\dots,A_k,A_l,\dots\}. \tag{6}$$

Then we use the selected nodes original feature weight to train the network and reuse other nodes which are not selected. We select the nodes after pre-training.The specific steps are as follows:

Extracting nodes features. In DBN, nodes effects by the forward and backward weights. We get the weight feature of each node from the weight matrix W. We denote the feature matrix of node k as A_k .

$$A_k=W_{\cdot k}. \tag{7}$$

Calculating node similarity. We get the node similarity of node k and node l by Eq. 8 . Other nodes similarities are calculated the same way.And in Eq. 9 , the dimension of A_k and A_l is m*n.

$$sim=\frac{\sum_{m}\sum_{n}((A_k)_{mn}\quad\overline{A_k})((A_l)_{mn}\quad\overline{A_l})}{\sqrt{(\sum_{m}\sum_{n}((A_k)_{mn}\quad\overline{A_k})^2)(\sum_{m}\sum_{n}((A_l)_{mn}\quad\overline{A_l})^2)}},\ \overline{A_k}=\text{mean}\,(A_k),\ \overline{A_l}=\text{mean}\,(A_l). \tag{8}$$

Selecting nodes. We select node similarity after the pre-training, and before the back propagation. Through Eq. 9, we get all node similarities, then we select nodes using the threshold C. For all nodes, if its similarity is less than C , its feature weight is remained; while if more than, only one of the similar nodes can be remained , other node feature weights get all-zero.

$$W'=\mu(W+\eta'\Delta W). \tag{9}$$

$$\mu = \begin{cases} 0 & \text{sim} > C \\ 1 & \text{sim} < C \end{cases}. \qquad (10)$$

Looping selecting. Compare the selected weight matrix with the original matrix,and set limit of the looping and the max-epoch.

Through above selection, we get the new weight matrix W’ , new matrix consists of two parts: the remaining node and the all-zero. In order to take full use of all the matrix ,we should reuse the all-zero matrix. Train again based on the new matrix. Through this method ,we get full use of all the nodes and the similarity between each two nodes is lower. The process of the improved DBN is in Fig. 2 .

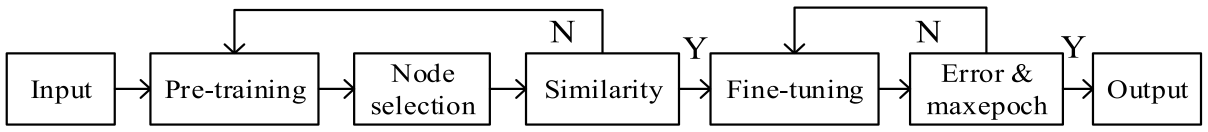

Fig. 2. The training process of improved DBN.

Experiments and analysis

We use MNIST data set to do experiments and a DBN network with two hidden layers, the number of input is 784 and the number of output is 10.We design three experiments. The details are as follows.

Experiment 1. In experiment 1, we do the pre-training and node selection once.After above steps, sum the numbers of nodes which act as the base vector. We change the numbers of the hidden layers nodes to check whether the nodes are excessive or not and when are excessive. Table 1 is the results.

As Table 1 shows, when the hidden layer nodes are increasing, the number of the selected nodes seem to be similar. The selected nodes number stands for the nodes with low similarity each other, in other words ,the selected nodes are of full use . Our proposed methods are meaningful.

Table 1. The remaining nodes of DBN with the different hidden nodes by selection.

Nodes number	Nodes selected	Rest nodes
[200 200]	45	[155 155]
[300 300]	117	[183 183]
[400 400]	218	[182 182]
[500 500]	323	[177 177]

Experiment 2. We calculate the similarity of the nodes after the nodes selection and the pre-training for more than one time. There are two hidden layers, and we get two figures.

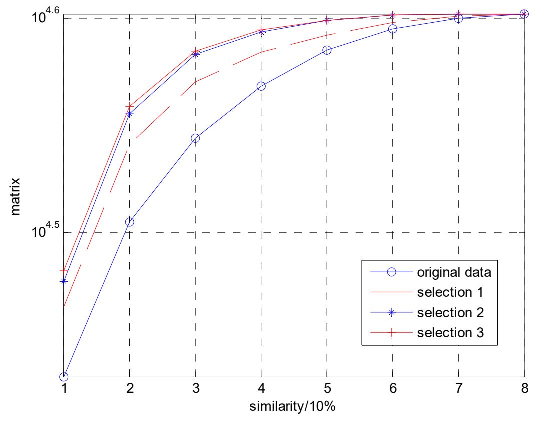

Fig. 3. The numbers of nodes under each similarity within the first hidden layer.

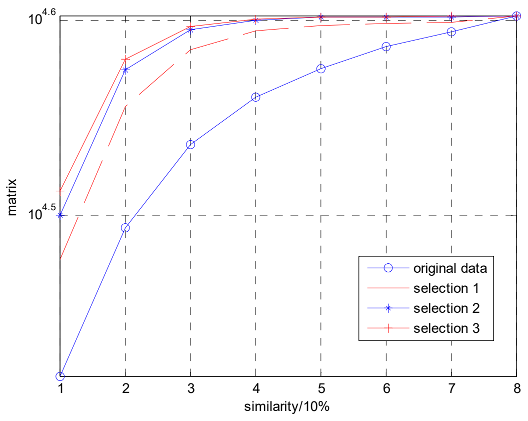

Fig. 4. The numbers of nodes under each similarity within the second hidden layer.

From the Fig. 3 and Fig. 4, we can get that the more times selections we do ,the less similar the nodes are. Fig. 3 shows, after three times of selection, almost all the nodes' similarity are under 50%, while the data in DBN without our proposed method is 80%. We reuse the all-zero nodes and train the network again, nodes get new weight matrix and the new matrix is not the same with the original matrix. That is to say, with selection all the nodes can be taken full use of and the selection is efficient.

Experiment 3. We train the whole DBN with our proposed method comparing the network without the method. The same with experiment 1 ,we change the numbers of hidden layer nodes .

Table 2. The accuracy of DBN without selection and DBN with selection.

The numbers of nodes	Accuracy/%	
	Without selection	With selection
[50 50]	89.72	91.65
[100 100]	92.54	92.65
[150 150]	93.45	94.24
[200 200]	93.08	94.68
[250 250]	93.72	94.84
[300 300]	93.47	94.77

As you can see in Table 2,the improved DBN gets the better result, no matter how many the nodes are. When the number of nodes is [50 50], it is clear that the nodes are not enough to train the network to get the high accuracy. Before we select nodes,there are still some nodes are similar , so the accuracy gets high when we reuse the nodes. Along with the increasing of the number of nodes ,there exists more nodes free. We reuse them and get the higher accuracy. The node selection method of DBN based on similarity gets better performance.

Conclusions

In order to make full use of all the hidden layer nodes, we propose a new method of node selection by comparing the similarity between each two nodes. Three experiments are put forward to verify the performance of the new method. And the results show that the new methods improve the performance. Our future work will be explored to adjust the parameters and the thresholds by self-adaption.

Acknowledgements

This work was financially supported by the Natural Science Foundation of Shandong Province (ZR2014FM039).

References

[1] Deng L. Deep Learning for Signal and Information Processing. Now Publishers, 2013.

[2] Hinton G E, Osindero S, Teh Y W. ,in: Neural Computation, 2006, 18(7):1527-1554.

[3] Tieleman T. Training restricted Boltzmann machines using approximations to the likelihood gradient[C]// International Conference. 2008:1064--1071.

[4] Desjardins G, Courville A C, Bengio Y, et al., in: Jmlr W & Cp Proc Aistats', 2010, 9:145-152.

[5] Subirats J L, Franco L, Jerez J M. C-Mantec, in: Neural Networks the Official Journal of the International Neural Network Society, 2011, 26(2):130-140.

[6] Zhang, C. Et al. 2013. ,in: Chinese Journal of Engineering Mathematics ,in Chinese, 2015(2),159-173.

[7] Grauman K, Fergus R. Learning Binary Hash Codes for Large-Scale Image Search[M]// Machine Learning for Computer Vision. Springer Berlin Heidelberg, 2013:49-87.

[8] Vidya R, Nasira D G M, Priyankka R P J. , in: Proceedings of the IEEE, 2014:124-127.