

Improved broad learning system: partial weights modification based on BP algorithm

Da Li¹, Shujuan Ji^{2*} and Chunjin Zhang³

¹ College of Computer Science and Technology, Shandong University of Science and Technology, Qingdao, Shandong 266590, China

² Key Laboratory for wisdom mine information technology of Shandong Province, Shandong University of Science and Technology, Qingdao, Shandong 266590, China

³ Network Information Center (NIC), Shandong University of Science and Technology, Qingdao, Shandong 266590, China

E-mail: jane_ji2003@aliyun.com

Abstract. Although breakthrough achievements of deep learning have been made in different areas, there is no good idea to prevent the time-consuming training process. Single-layer feedforward neural networks (e.g. BLS) are used to reduce the training time. However, with the decrease of training time, the accuracy degradation has emerged. In view of the limitation of random generation of connection parameters between feature nodes and enhancement nodes, this paper presents an algorithm (IBLS) based on BLS and backpropagation algorithm to learn the weights between feature nodes and enhancement nodes. Experiments over NORB and MNIST data sets show that the improved broad learning system achieves acceptable results.

1. Introduction

Recently, neural networks, especially deep learning (deep neural networks), have become a hot topic in machine learning field. Breakthrough achievements have been made in different areas such as computer vision, speech recognition, and natural language processing. Especially, convolutional neural networks (CNN) [1-2], deep belief networks (DBN) [3-4], and deep Boltzmann machines (DBM) [5] are widely used in above areas. However, these neural networks have many layers, thus the number of the connection parameters and hyper-parameters are very large. As a result, the training process is very time-consuming.

Single-hidden layer feedforward neural networks is the simplest neural network. It is one of the most widely used and fastest growing artificial neural networks. The traditional training method is a gradient-based learning algorithm [6]. This method has the disadvantage of being sensitive to parameters and falling into a local optimal solution. In order to solve this problem, Pao [7-9] proposed a new method named random vector functional link neural network (RVFLNN). Except for solving the problem of parameters sensitivity and a local optimal solution, this method also overcomes the shortcoming of long training time.

With the development of economy, the amount as well as dimensions [10] of data keeps increasing. Directly taking the original data to the network will make the network unable to work properly. So, Chen et al. [11] proposed a broad learning system (BLS) based on RVFLNN.

BLS has effectively solved the problem of large volumes and high dimensions of data, but meanwhile caused the lower classification accuracy. In this paper, an improved broad learning system



(IBLS) is developed. In IBLS, we modify the randomly generated weights between feature nodes and enhancement nodes using backpropagation algorithm to achieve the purpose of improving the accuracy.

2. Theory

The broad learning system (BLS) [11] is designed based on RVFLNN. The difference between BLS and RVFLNN is that the latter model directly takes the raw data as input, while the former model maps the raw data into mapped features before entering the network. The structure of BLS is designed as follows. First, the raw data is mapped into mapped features. Then, the mapped features are enhanced as enhancement nodes with randomly generated weights. Finally, all mapped features and enhancement nodes are connected to the output. The network structure of BLS is shown in figure 1.

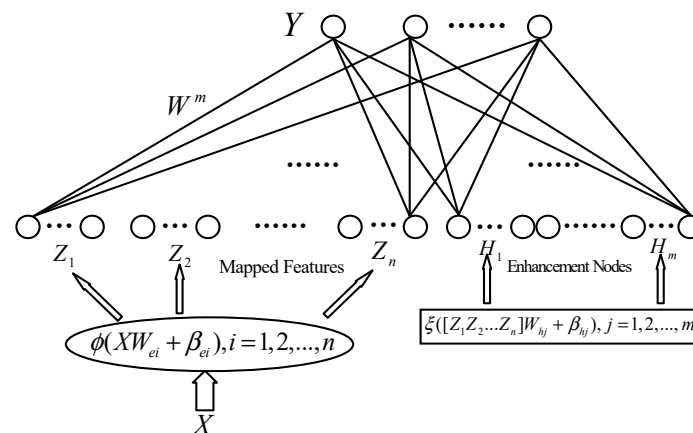


Figure 1. Broad learning system.

Suppose that we denote the input data as X and project the data, using $\phi_i(XW_{ei} + \beta_{ei})$, to become the i th mapped feature, Z_i . Similarly, the j th enhancement nodes, $\xi_j(Z^i W_{hj} + \beta_{hj})$ is denoted as H_j , where $Z^i = [Z_1, Z_2, \dots, Z_i]$. When i or j is assigned with different values, the mapping functions ϕ_i may not be the same.

Therefore, the structure of BLS can be represented by the following equation

$$\begin{aligned} Y &= [Z_1, \dots, Z_n | \xi(Z^n W_{h1} + \beta_{h1}), \dots, \xi(Z^n W_{hm} + \beta_{hm})] W^m \\ &= [Z_1, \dots, Z_n | H_1, \dots, H_m] W^m \\ &= [Z^n | H^m] W^m \end{aligned} \quad (1)$$

where W^m is the weight matrix connected to the output layer.

Solve above equation, we get that

$$\begin{aligned} W^m &= [Z^n | H^m]^+ Y \\ &= \lim_{\lambda \rightarrow 0} ([Z^n | H^m]^T [Z^n | H^m] + \lambda I)^{-1} [Z^n | H^m]^T Y \end{aligned} \quad (2)$$

3. The model

In BLS, the weights between feature nodes and enhancement nodes are randomly generated and fixed. Although it can avoid the burden of modifying weights, there will be some randomness, which will affect the final classification accuracy. In order to reduce the randomness and improve the classification accuracy, we combine the backpropagation algorithm with the broad learning system, that is, use the backpropagation algorithm to modify the randomly generated weights. The structure of proposed model is shown in figure 2.

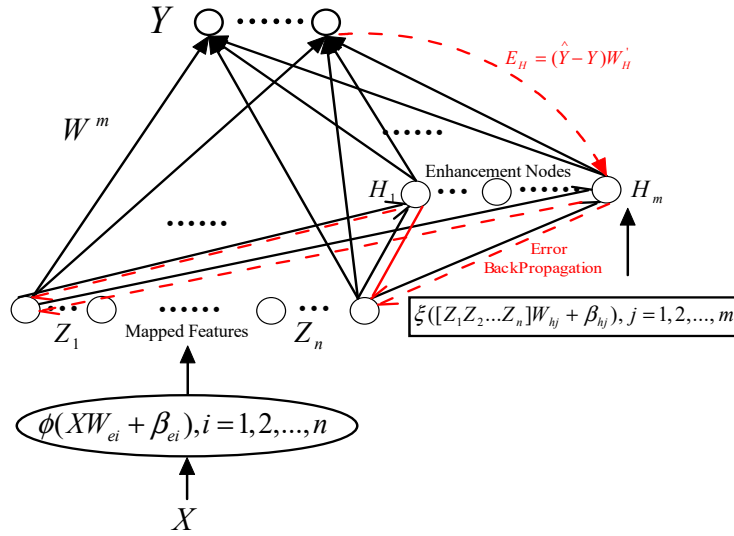


Figure 2. The structure of proposed model.

The structure of the model is design as follows. First, the mapped features are generated from the raw data to form the feature nodes using sparse auto-encoder model [12-14] (see steps (1) - (5)). Second, enhancement nodes are obtained from feature nodes with randomly generated weights (see steps (6) - (10)). Then calculate the weights connected to the output layer (see steps (11)). Finally, calculate the gradient of enhancement nodes and modify the weights and biases between feature nodes and enhancement nodes using backpropagation algorithm (see steps (12) - (17)). Algorithm 1 gives the detailed training process, the time complexity of the algorithm is $O(n + m + t)$, where n is the number of feature nodes, m is the number of enhancement nodes, and t denotes the number of iterations when the algorithm converges.

Algorithm 1: Improved broad learning system algorithm.

Input: training set X

Output: W

- (1) for $i=0:n$ do
 - (2) Calculate W_e using sparse auto-encoder model;
 - (3) Calculate $Z_i = [\phi(XW_{ei} + \beta_{ei})]$, β_{ei} is randomly generated;
 - (4) end
 - (5) Set the feature mapping group $Z^n = [Z_1, Z_2, \dots, Z_n]$;
 - (6) for $j=1:m$ do
 - (7) Random W_{hj}, β_{hj} ;
 - (8) Calculate $H_j = [\xi(Z^n W_{hj} + \beta_{hj})]$;
 - (9) end
 - (10) Set the enhancement nodes group $H^m = [H_1, H_2, \dots, H_m]$;
 - (11) Set $A^m = [Z^n | H^m]$ and calculate $(A^m)^+$;
 - (12) while the terminated conditions is not satisfied do
 - (13) Calculate output of the network;
 - (14) Calculate errors of enhancement nodes layer using $E_H = (\hat{Y} - Y)W'_H$;
 - (15) Calculate gradient of enhancement nodes;
 - (16) Update W_{hj} and β_{hj} ;
 - (17) end
 - (18) return W
-

4. Experiments and analysis

In this section, we compare the classification ability of our method with existing mainstream methods, including Stacked Auto Encoders (SAE) [4], another version of stacked auto-encoder (SDA) [15], Deep Belief Networks (DBN) [3], Multilayer Perceptron-based methods (MLP) [16], Deep Boltzmann Machines (DBM) [5] and the original method (BLS), to verify the advantages of proposed method. As we know, the improvement on the BLS is just applying the system to other fields and methods [17-19], so there is no comparison with them.

4.1. Data sets

These experiments are applied on popular small NYU Object Recognition Benchmark (V1.0) [20] and MNIST data set [6]. NORB database contains images of 50 toys belonging to 5 generic categories: four-legged animals, human figures, airplanes, trucks, and cars. Each picture consists of $2 \times 32 \times 32$ pixels, as shown in figure 3. The training set consists of five instances of each category (instances 4, 6, 7, 8 and 9) for a total of 24,300 images, the test set consists of the remaining five instances (instances 0, 1, 2, 3 and 5) for a total of 24,300 images.

MNIST dataset contains 70,000 handwritten digital images, each of which is a 28×28 gray-scaled image. Of these, 60,000 images are used as training set and 10,000 images are used as test set. Samples of hand-written digital images are shown in figure 4.

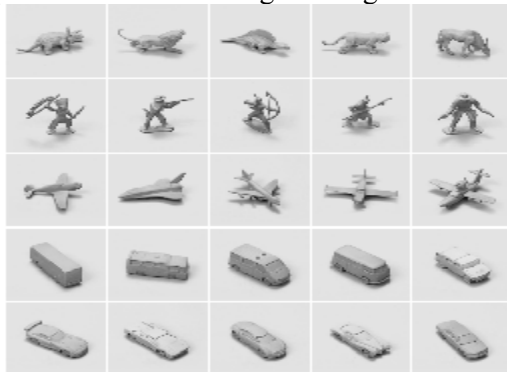


Figure 3. Samples of NORB data set.



Figure 4. Samples of MNIST data set.

4.2. Experimental environment and settings

The original method and proposed method have the same structure (100 feature nodes and 11000 enhancement nodes) for MNIST data set. Meanwhile, the deep structures of SAE, DBN, and DBM is 1000-500-25-30, 500-500-2000 and 500-500-1000, respectively. For NORB data set, the structure of proposed method, as same as the original method, has 100×10 features nodes, 1×9000 enhancement nodes. The deep and complex structure of DBN is 4000-4000-4000. The above experiments are tested on Matlab software platform under a PC that equips with Intel-i7 2.4 GHz CPU, 16G memory.

4.3. Performance comparison and analysis

The accuracy and training time of the original algorithm and proposed method as well as the existing state-of-the-art on NORB and MNIST data sets are shown in table 1 and table 2, respectively.

Table 1. Accuracy and training time on NORB data set.

Method	Accuracy/%	Training time/s
SAE	86.28	60504.34
SDA	87.62	65747.69
DBN	88.47	87280.42
DBM	89.65	182183.53
MLP	84.20	34005.470
BLS	89.27	41.4666
IBLS	90.32	66.1383

From Table 2, we can observe that our proposed method (IBLS) gains the highest accuracy 90.32%. The modification of the weights which is randomly generated in BLS plays an import role in improving the accuracy. For training time in table 2, the improved broad learning system is 66.1383 seconds. It is much faster than other methods (excluding BLS), in fact, just a little slower than BLS. For small data sets, the proposed method can achieve the best accuracy of classification with little extra time.

Table 2. Accuracy and training time on MNIST data set.

Method	Accuracy/%	Training time/s
SAE	98.60	36448.40
SDA	98.72	37786.03
DBN	98.87	53219.77
DBM	99.05	121455.69
MLP	97.39	21468.12
BLS	98.74	78.68
IBLS	98.87	3047.57

From the table 3, we can observe that the accuracy of the proposed model IBLS gets 98.87%, which is better than SAE, SDA, MLP and BLS. The improvement of the accuracy is due to the modified parameters. Modification of weights can avoid randomness, and achieve better combination of mapped features to form enhancement nodes. For training time, the improved broad learning system is 3047.57 seconds. It is not the best one, but it is still faster than SAE, SDA, DBM, DBM and MLP. The reason why the training time is higher than BLS is mainly because of the memory's capacity. Since the pseudo-inverse matrix is already stored in the memory, the process of calculating the gradient will make the amount of data larger than the memory's capacity, therefore it is necessary to switch the data while calculating. Data switching in memory is a time-consuming process, so the training time is longer than those methods. For large data sets, the proposed method is a compromised method between accuracy and time.

5. Conclusion

The improved broad learning system (IBLS) is developed in this paper, which aims to improve the accuracy of the original algorithm. This method is a combination of backpropagation algorithm and broad learning system, namely, the randomly generated weights from the feature nodes to the enhancement nodes are modified using the backpropagation algorithm. Modification of parameters could prevent randomness, and achieve better combination of mapped features to form enhancement nodes, as a result, the classification is more accurate. Acceptable results on NORB and MNIST data sets are obtained of the proposed method.

However, the improved broad learning system is for static and mixed data sets. In the future, we will improve the model to adapt incremental and cross-domain classification data sets which are common in real life.

Acknowledgments

This paper is supported in part by the Natural Science Foundation of China (No. 71772107, 71403151, 61502281), the Natural Science Foundation of Shandong Province (Nos. ZR2013FM023, ZR2014FP011), Shandong Education Quality Improvement Plan for Postgraduate, China's Postdoctoral Science Fund (No. 2014M561948), Postdoctoral innovation project special funds of Shandong Province (No. 201403007), Applied research project for Qingdao postdoctoral researcher, Project of Shandong Province Higher Educational Science and Technology Program (J14LN33), the Leading talent development program of Shandong University of Science and Technology and special funding for Taishan Scholar construction project.

References

- [1] Lecun Y, Bottou L, Bengio Y and Haffner P 1998 *Proceedings of the IEEE* **86** 2278-324
- [2] Sarwar S S, Ankit A and Roy K 2017 arXiv:1712.02719v1 [cs.CV]

- [3] Hinton G E, Osindero S and Teh Y W 2014 *Neural Computation* **18** 1527-54
- [4] Hinton G E and Salakhutdinov R R 2006 *Science* **313** 504-7
- [5] Salakhutdinov R and Hinton G E 2009 *Journal of Machine Learning Research* **5** 1967-2006
- [6] LeCun Y and Boser B 1990 *Advances in Neural Information Processing Systems* **2** 396-404
- [7] Pao Y H and Takefuji Y 1992 *Computer* **25** 76-9
- [8] Pao Y H, Park G H and Sobajic D J 1994 *Neurocomputing* **6** 163-80
- [9] Igel'nik B and Pao Y H 1995 *IEEE Transactions on Neural Networks* **6** 1320
- [10] Chen C L P and Zhang C Y 2014 *Information Sciences* **275** 314-47
- [11] Chen C L P and Liu Z 2017 *IEEE Transactions on Neural Networks & Learning Systems* **29** 1-15
- [12] Gong M, Liu J, Li H, Cai Q and Su L 2015 *IEEE Transactions on Neural Networks & Learning Systems* **26** 3263
- [13] Xu Z, Chang X, Xu F and Zhang H 2012 *IEEE Transactions on Neural Networks & Learning Systems* **23** 1013
- [14] Yang W, Gao Y, Shi Y, Cao L 2015 *IEEE Transactions on Neural Networks & Learning Systems* **26** 2801-15
- [15] Vincent P, Larochelle H, Bengio Y and Manzagol P A 2008 *Proceedings of the 25th international conference on Machine learning* 1096-103
- [16] Bishop and Christopher M 2006 *Pattern Recognition and Machine Learning* (New York: Springer)
- [17] He W, Yan Z, Sun C and Chen Y 2017 *IEEE Transactions on Cybernetics* **47** 3452-65
- [18] Liu Z and Chen C L P 2017 *International Conference on Security, Pattern Analysis, and Cybernetics* 136-41
- [19] Liu Z, Zhou J and Chen C. L. P 2017 *International Conference on Information, Cybernetics and Computational Social Systems* 683-7
- [20] Lecun Y, Huang F J and Bottou L 2004 *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004* **2** 97-104