Dense Broad Learning System based on Conjugate Gradient

1st Weidong Zou
School of Automation
Beijing Institute of Technology
Beijing, China
zouweidong1985@163.com

2nd Yuanqing Xia
School of Automation
Beijing Institute of Technology
Beijing, China
xia_yuanqing@bit.edu.cn

3rd Weipeng Cao*
College of Computer Science and Software Engineering
Shenzhen University
Shenzhen, China
caoweipeng@szu.edu.cn

***Abstract*—Conventional training mechanism for deep learning, which is based on gradient descent, suffers from many notorious issues such as low convergence rate, over-fitting, and time-consuming. To alleviate these problems, a novel deep learning algorithm with a different learning mechanism named Broad Learning System (BLS) was proposed by Prof. C. L. Philip Chen in 2017. BLS randomly selects the parameters of the feature nodes and enhancement nodes during its training process and uses the ridge regression theory to solve its output weights. BLS has been widely used in many fields because of its high efficiency. However, there is a fundamental problem that has not yet been solved, that is, the appropriate value of the parameter λ for the ridge regression operation of BLS is difficult to be set properly, which often leads to the problem of over-fitting and seriously limits the development of BLS. To solve this problem, we proposed a novel Dense BLS based on Conjugate Gradient (CG-DBLS) in this paper, in which each feature node is connected to other feature nodes and each enhancement node is connected to other enhancement nodes in a feed-forward fashion. The recursive least square method and conjugate gradient method are used to calculate the output weights of the feature nodes and enhancement nodes respectively. Experiment studies on four benchmark regression problems from UCI repository show that CG-DBLS can achieve much lower error and much higher stability than BLS and its variants.**

***Index Terms*—broad learning system, conjugate gradient, neural networks with random weights, random vector functional link neural network**

I. Introduction

In recent years, deep learning based architectures, notably Convolutional Neural Network (CNN) and Restricted Boltzmann Machines (RBM), have enabled rapid advances in the field of artificial intelligence in general [3]. This progress has triggered many real-world applications such as face detection and recognition [12], blur image identification [16], mandible segmentation [18], and speaker recognition [9]. But most traditional deep learning algorithms are based on the gradient descent method for model training, which suffers from many disadvantages such as time-consuming.

To alleviate these problems and improve the generalization ability of neural networks, Prof. C. L. Philip Chen et al. proposed the Broad Learning System (BLS) in 2017 [4], which is based on the notion of Random Vector Functional Link neural network (RVFL) [2], [15]. BLS provides a simple and efficient method to train a flat network with randomized parameters. Specifically, in BLS, the input weights of the hidden layer nodes (both the feature nodes and the enhanced nodes) are assigned randomly from a given range and the output weights are calculated by using the ridge regression theory. The learning process of BLS is non-iterative, which is different from the traditional deep learning methods. This characteristic makes BLS have very fast learning speed than the traditional neural networks (i.e., neural networks trained by the gradient descent algorithm). Up to now, there are many BLS based algorithms have been proposed and widely used to solve the real-life tasks [6], [13], [17], [20], [21]. For example, to improve the modeling ability of BLS on uncertain data, Jin et al. [10] proposed Regularized robust BLS (RBLS), which assumes that the output weight and regression residual error follow Laplacian error distribution. Jin et al. proposed a new objection function for RBLS and used a new method to calculate the output weights of the system. Later, Feng et al. found that using the Takagi-Sugeno (TS) fuzzy system and sparse auto-encoder can further compact the network architecture of BLS and the number of the fuzzy rule can be determined by k-means method [7].

Although BLS and its variants have shown great application potential, there are still many basic problems that have not been solved due to the relevant research just started. For example, the regularization parameter λ is an indispensable parameter for BLS based algorithms, but there is still no effective method to choose an appropriate value for it in a specific learning system. The existing BLS based algorithms set the value of λ through the trial and error method, which is very clumsy and time-consuming. What is worse, because the generalization performance of BLS is relatively sensitive to λ, so most of the BLS based algorithms are not very stable.

To solve this problem, we proposed a novel Dense BLS based on Conjugate Gradient (CG-DBLS) in this paper. The original idea of CG-DBLS is to connect each feature node to other feature nodes and connect each enhancement node

This work was supported by the National Key Research and Development Program of China (2018YFB1003700), National Natural Science Foundation of China (61906015, 61836001, 61672358), Opening Project of Shanghai Trusted Industrial Control Platform (TICPSH202003008-ZC), and Guangdong Science and Technology Department (Grant no. 2018B010107004).

*Corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

to other enhancement nodes in a feed-forward fashion, and then calculate the output weights of the feature nodes and enhancement nodes via the recursive least square method and the conjugate gradient method respectively. CG-DBLS can avoid manually selecting the value of λ, thereby improving the stability and generalization ability of the model. We evaluate the performance of CG-DBLS on four benchmark data sets from the UCI repository and the experimental results show that CG-DBLS significantly outperform BLS and fuzzy BLS on all the case.

The contributions of this study can be summarized as follows.

(1) We introduced a new dense architecture for BLS and showed that the efficiency of this architecture is much higher than that of the original BLS and its variants such as FBLS [7].

(2) The recursive least square method and conjugate gradient method were applied to solve the output weights of BLS for the first time, which provides researchers with a new perspective to improve BLS.

(3) This paper presents an example of the integration of dense BLS, recursive least square method, and conjugate gradient method, which can inspire researchers to design new deep architecture for BLS.

The remaining of this paper is organized as follows. In Sec. II, we briefly introduce the learning mechanisms of BLS and FBLS. The conjugate gradient method, the proposed CG-DBLS and its pseudo-code are presented in Sec. III. In Sec. IV, we show the simulation experiments and the corresponding analysis. In Sec. V, we conclude this study.

II. PRELIMINARIES

A. Broad Learning System

Given a training data set $\{\mathbf{X}, \mathbf{Y}\} \in \Re^{(d+t)\times N}$, where d denotes the feature dimension of the input data, t denotes the number of the class in classification problem, and N denotes the number of the training data. The activation function of feature nodes and enhancement nodes in the Broad Learning System (BLS) is piecewise and continuous. The model of BLS with n feature nodes and m enhancement nodes is generally expressed as a linear combination of functions as follows.

$$\mathbf{y}_{n+m}(\mathbf{X}) = \sum_{i=1}^{n} \omega_i \mathbf{z}_i + \sum_{j=1}^{m} \omega_{n+j} \mathbf{h}_j \tag{1}$$

$$\mathbf{z}_i = \phi(\mathbf{a}_{ei} \cdot \mathbf{X} + b_{ei}), i = 1, 2, ..., n, \mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n], \tag{2}$$

$$\mathbf{h}_j = \psi(\mathbf{a}_{hj} \cdot \mathbf{z} + b_{hj}), j = 1, 2, ..., m, \mathbf{h} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_m], \tag{3}$$

where

$\mathbf{z}$ refers to the outputs of the feature nodes,

$\mathbf{h}$ refers to the outputs of enhancement nodes,

$\mathbf{a}_{ei}$ is the weight vector connecting the input layer to the ith feature node,

b_{ei} is the bias parameter of the ith feature node,

$\mathbf{a}_{hj}$ is the weight vector connecting the feature node to the jth enhancement node,

b_{hj} is the bias parameter of the jth enhancement node,

ω_i is the weight connecting the ith feature node to the output layer,

and ω_{n+j} is the weight connecting the jth enhancement node to the output layer.

The output weight of BLS can be computed by using the ridge regression approximation of pseudo-inverse

$$\mathbf{W}_{BLS} = [\omega_1, ..., \omega_{n+m}] = [\mathbf{z}, \mathbf{h}]^{\dagger} \cdot \mathbf{Y} \tag{4}$$

The universal approximation of BLS has been proven by using lemma II-A.

Lemma II-A [5] Given arbitrary compact set $\mathbf{M} \subset \mathbf{I}^d$ and arbitrary continuous target function $\mathbf{f} \in C(\mathbf{I}^d)$, there is a progression $\mathbf{y}_{n+m}$ in BLS with n feature nodes and m enhancement nodes, and a respective progression of probability measures τ_{n+m}, such that

$$\lim_{m,n\to\infty} \rho_{\mathbf{M}}(\mathbf{f}, \mathbf{y}_{n+m}) = \lim_{m,n\to\infty} \sqrt{E[\int_{\mathbf{M}} (\mathbf{f} - \mathbf{y}_{n+m})^2]} = 0. \tag{5}$$

B. Fuzzy Broad Learning System

To enhance the robustness of BLS, Feng et al. proposed Fuzzy BLS (FBLS) [7]. The difference between FBLS and BLS is that, in FBLS, the feature nodes are replaced by a set of Takagi-Sugeno (TS) fuzzy subsystems. And K-Means algorithm is used to block sort the input data, optimize the centers of Gaussian membership functions and the number of fuzzy rules for the fuzzy subsystems.

Given a training data set $\{\mathbf{X}, \mathbf{Y}\} \in \Re^{(d+t)\times N}$, l fuzzy subsystems, and p enhancement nodes, the model of FBLS can be expressed as follow:

$$\mathbf{y}_{l+p}(\mathbf{X}) = \sum_{i=1}^{l} \nu_i \mathbf{V}_i \boldsymbol{\varphi}_i + \sum_{j=1}^{p} \gamma_j \mathbf{u}_j = \sum_{i=1}^{l} \nu_i \mathbf{r}_i + \sum_{j=1}^{p} \gamma_j \mathbf{u}_j \tag{6}$$

$$\begin{gathered} \mathbf{V}_i = diag[\sum_{t=1}^{M} \vartheta_{kt}^i x_{1t}, ..., \sum_{t=1}^{M} \vartheta_{kt}^i x_{Nt}], \\ \boldsymbol{\varphi}_i = [\boldsymbol{\omega}_1^i, ..., \boldsymbol{\omega}_q^i, ..., \boldsymbol{\omega}_{K_i}^i], \boldsymbol{\omega}_q^i = [\omega_{1q}^i, ..., \omega_{K_{Nq}}^i] \end{gathered} \tag{7}$$

$$\mathbf{r}_i = \mathbf{V}_i \boldsymbol{\varphi}_i, i = 1, 2, ..., l, \mathbf{r} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_l], \tag{8}$$

$$\mathbf{u}_j = \psi(\mathbf{a}_{Fj} \cdot \mathbf{r} + b_{Fj}), j = 1, 2, ..., p, \mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_p], \tag{9}$$

where

$\mathbf{r}$ refers to the outputs of the fuzzy subsystems,

$\mathbf{u}$ refers to the outputs of the enhancement nodes,

$\mathbf{a}_{Fj}$ is the weight vector connecting the fuzzy subsystems to the jth enhancement node,

b_{Fj} is the bias parameter of the jth enhancement node,

ϑ_{kt}^i is the coefficient of the ith fuzzy sets,

and $\boldsymbol{\varphi}_i$ is the weighted fire strength for all fuzzy rules of the ith fuzzy subsystem.

The output weights of FBLS are computed by using the ridge regression approximation of pseudo-inverse

$$\mathbf{W}_{FBLS} = [\nu_1, ...\nu_l, \gamma_1, ..., \gamma_p] = [\mathbf{r}, \mathbf{u}]^{\dagger} \cdot \mathbf{Y} \tag{10}$$

III. DENSE BROAD LEARNING SYSTEM BASED ON CONJUGATE GRADIENT

To better introduce our method, we present this section as the following steps. Firstly, we introduce the conjugate gradient method in section III-A. Secondly, the details of the proposed Dense Broad Learning System based on Conjugate Gradient (CG-DBLS) are given in section III-B. Finally, we share the pseudo-code of CG-DBLS in section III-C.

A. *Conjugate Gradient Method*

Conjugate gradient method (CG) [11] is one of the effective methods for solving the linear system of equations. The following iteration process is used to generate an approximation sequence.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, k = 0, 1, 2, ... \tag{11}$$

where $\mathbf{x}_k$ refers to the point of current iteration, the step-size α_k is positive number, and the search direction $\mathbf{d}_k$ is represented as:

$$\mathbf{d}_k = \begin{cases} -\mathbf{g}_k, k = 0 \\ -\mathbf{g}_k + \beta_k^{FR}\mathbf{d}_{k-1}, k \geq 1 \end{cases} \tag{12}$$

where $\mathbf{g}_k$ is the gradient, and the conjugate parameter β_k^{FR} [8] can be represented as:

$$\beta_k^{FR} = \frac{\|\mathbf{g}_k\|^2}{\|\mathbf{g}_{k-1}\|^2}. \tag{13}$$

B. *Dense Broad Learning System based on Conjugate Gradient (CG-DBLS)*

The network structure of CG-DBLS is shown in Fig. 1. Given a training data set $\{\mathbf{X}, \mathbf{Y}\} \in \Re^{(d+t)\times N}$, the activation function of the feature nodes and enhancement nodes in CG-DBLS is required to be piecewise and continuous, the model of CG-DBLS with n feature nodes and m enhancement nodes can be expressed as a linear combination of functions.

$$\mathbf{y}_{n+m}(\mathbf{X}) = \sum_{i=1}^{n} \rho_i \tilde{\mathbf{Z}}_i + \sum_{j=1}^{m} \delta_j \tilde{\mathbf{H}}_j, \tag{14}$$

$$\begin{aligned} \mathbf{Z}_i &= \phi(\mathbf{a}_{ei} \cdot \mathbf{X} + b_{ei}), \varepsilon_{Fi} = \frac{\|\mathbf{Z}_i\|^2}{1 + \|\mathbf{Z}_i\|}, \\ \tilde{\mathbf{Z}} &= [\mathbf{Z}_1, \mathbf{Z}_2 + \varepsilon_{F1}\mathbf{Z}_1, ..., \mathbf{Z}_n + \sum_{p=1}^{n-1} \varepsilon_{Fp}\mathbf{Z}_p], \end{aligned} \tag{15}$$

$$\begin{aligned} \mathbf{H}_j &= \psi(\mathbf{a}_{hj} \cdot \tilde{\mathbf{Z}} + b_{hj}), \varepsilon_{Ej} = \frac{\|\mathbf{H}_j\|^2}{1 + \|\mathbf{H}_j\|}, \\ \tilde{\mathbf{H}} &= [\mathbf{H}_1, \mathbf{H}_2 + \varepsilon_{E1}\mathbf{H}_1, ..., \mathbf{H}_m + \sum_{p=1}^{m-1} \varepsilon_{Ep}\mathbf{H}_p], \end{aligned} \tag{16}$$

where

ε_{Fi} refers to the appropriate coefficients of the ith feature node,

ε_{Ej} refers to the appropriate coefficients of the jth enhancement node,

TABLE I

DETAILS OF THE FOUR REGRESSION DATA SETS.

Dataset	Instance	Attributes
Airfoil Self-Noise	1503	6
Combined Cycle Power Plant	9568	4
Concrete Compressive Strength	1030	9
Energy efficiency	768	8

ρ_i is the output weight of the ith feature node, which is calculated by using the recursive least square method,

and δ_j is the output weight of the jth enhancement node, which is calculated by using the conjugate gradient method.

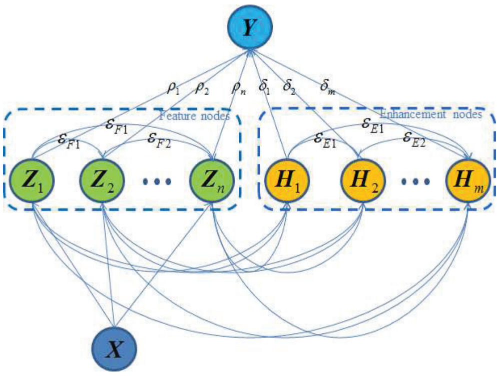

Fig. 1. Topological structure of CG-DBLS.

C. *Pseudo-code of CG-DBLS*

The pseudo-code of the proposed CG-DBLS can be summarized as algorithm 1.

Next, we test the performance of the CG-DBLS on four benchmark data sets.

IV. SIMULATION EXPERIMENTS AND DISCUSSIONS

Four regression problems from the UCI repository were used to test the generalization ability of CG-DBLS in this paper. The details of the four regression data sets are shown in table I. All the experiments were conducted in PC with Intel(R) Core(TM) i7-3520M CPU with NVIDIA NVS 5400M and 8GB RAM.

The performance statistical indexes include Root Mean Square Error (RMSE), Ratio of Standard Deviation (RSD) [19], and Mean of Percent Error (MPE) [14]. They can be calculated using the following equations:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}(u_i - \hat{u}_i)^2}{m}}, \tag{17}$$

$$RSD = \sqrt{\frac{\sum_{i=1}^{m}(u_i - \tilde{u})^2}{\sum_{i=1}^{m}(\hat{u}_i - \bar{u})^2}}, \tag{18}$$

Algorithm 1 CG-DBLS algorithm

Input: A training data set $\{\mathbf{X}, \mathbf{Y}\} \subset \mathbf{R}^n \times \mathbf{R}$.

Output: All the parameters of the CG-DBLS model.

```
Initialization: set $\mathbf{e}_0 = \mathbf{Y}$.
for $p = 1; p \leq n$ do
    Randomly assign the parameters connecting the input layer to the feature node $(\mathbf{a}_{ep}, b_{ep})$;
    Calculate the output of the feature node $\mathbf{Z}_p = [\phi(\mathbf{a}_{ep} \cdot \mathbf{X} + b_{ep})]$;
    Calculate the appropriate coefficient of the $p$-th feature node $\varepsilon_{Fp} = \frac{\|\mathbf{Z}_p\|^2}{1+\|\mathbf{Z}_p\|}$;
end for
Calculate the output matrix of the feature nodes for CG-DBLS $\tilde{\mathbf{Z}} = [\mathbf{Z}_1, ..., \mathbf{Z}_n + \sum_{i=1}^{n-1} \varepsilon_{Fi}\mathbf{Z}_i]$;
for $k = 1; k \leq m$ do
    Randomly assign the parameters connecting the feature node to the enhancement node $(\mathbf{a}_{hk}, b_{hk})$;
    Calculate the output of the enhancement node $\mathbf{H}_k = [\phi(\mathbf{a}_{hk} \cdot \tilde{\mathbf{Z}} + b_{hk})]$;
    Calculate the appropriate coefficient of the $k$-th enhancement node $\varepsilon_{Ek} = \frac{\|\mathbf{H}_k\|^2}{1+\|\mathbf{H}_k\|}$;
end for
Calculate the output matrix of the enhancement nodes for CG-DBLS $\tilde{\mathbf{H}} = [\mathbf{H}_1, ..., \mathbf{H}_m + \sum_{j=1}^{m-1} \varepsilon_{Ej}\mathbf{H}_j]$
for $q = 1; q \leq n$ do
    Calculate the output weight of the $q$-th feature node $\rho_q = \frac{\langle \mathbf{e}_{q-1}, \tilde{\mathbf{Z}}_q \rangle}{\|\tilde{\mathbf{Z}}_q\|^2}$;
    Calculate the network output error of CG-DBLS with $q$ feature nodes $\mathbf{e}_q = \mathbf{Y} - \sum_{i=1}^{q} \rho_i \tilde{\mathbf{Z}}_i$;
end for
Calculate the network output error of CG-DBLS with $n$ feature nodes $\mathbf{E} = \mathbf{Y} - \sum_{i=1}^{n} \rho_i \tilde{\mathbf{Z}}_i$;
Randomly assign the parameters connecting the enhancement node to the output layer $\mathbf{W}_{E0} = [\delta_1, ..., \delta_m]$;
Given $k = 0, \epsilon > 0$, and calculate the error $\mathbf{r}_0 = (\tilde{\mathbf{H}})^T \cdot \mathbf{E} - (\tilde{\mathbf{H}})^T \cdot \tilde{\mathbf{H}} \cdot \mathbf{W}_{E0}$;
while $\|\mathbf{r}_k\| > \epsilon$ do
    $k = k + 1$;
    if $k = 1$ then
        $\mathbf{p}_1 = \mathbf{r}_0$;
    else
        $\mathbf{p}_k = \mathbf{r}_{k-1} + \frac{(\mathbf{r}_{k-1})^T \cdot \mathbf{r}_{k-1}}{(\mathbf{r}_{k-2})^T \cdot \mathbf{r}_{k-2}} \mathbf{p}_{k-1}$;
    end if
end while
$\alpha_k = \frac{(\mathbf{r}_{k-1})^T \cdot \mathbf{r}_{k-1}}{(\mathbf{p}_k)^T \cdot (\tilde{\mathbf{H}})^T \cdot \tilde{\mathbf{H}} \cdot \mathbf{p}_k}$,
$\mathbf{W}_{Ek} = \mathbf{W}_{E(k-1)} + \alpha_k \mathbf{p}_k$,
$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k (\tilde{\mathbf{H}})^T \cdot \tilde{\mathbf{H}} \cdot \mathbf{p}_k$;
Set the output weight of the enhancement node to $\mathbf{W}_{Ek}$.
return All the parameters of the neural network.
```

$$MPE = \frac{1}{m} \sum_{i=1}^{m} |\frac{u_i - \hat{u}_i}{\hat{u}_i}| \times 100\%, \tag{19}$$

where

u_i refers to the predicted value,

$\tilde{u}$ refers to the average of the predicted values,

$\hat{u}_i$ refers to the true value,

$\bar{u}$ refers to the average of the true values,

and m denotes the number of the samples.

Note that the smaller the values of RMSE and MPE, the better the prediction performance of the model; the larger the value of RSD, the better the stability of the model.

To verify the efficiency of our method, we compared the performance of CG-DBLS with BLS and FBLS on these data sets. In our experiments, BLS, FBLS and CG-DBLS recruit the parameters of enhancement nodes randomly from $[-1, 1]^d \times [-1, 1]$, which are based on a uniform sampling distribution. The enhancement nodes of BLS, FBLS, and CG-DBLS are a hyperbolic tangent function of additive nodes. The number of the enhancement node in BLS, FBLS and CG-DBLS are determined from $[40, 50, 60, 70, 80, 90, 100, 110, 120, 130]$, and the number of the feature node in BLS, and CG-DBLS is set to 25. The regularization parameter of BLS, FBLS, and CG-DBLS is $\lambda = 15000$. In addition, FBLS includes 25 fuzzy subsystems, and the number of the rules in each fuzzy subsystem is 2.

The experimental procedure of the 10-fold cross-validation is as follows:

(1) The order of the samples is disrupted randomly;

(2) All the samples are divided equally into ten parts;

(3) Nine parts of the samples are chosen as the training data, and the remaining part is set to be the testing data;

(4) Next, the step (3) is repeatedly run ten times, so one can obtain ten RMSEs, RSDs, and MPEs from the experiments;

(5) The mean value of the ten RMSEs, RSDs, and MPEs is the 10-fold cross-validation value.

The experimental results are shown in figures 2-5. Take figure 2 as an example, from figure 2 we can observe that the values of RMSE and MPE of the proposed CG-DBLS are much smaller than that of BLS and FBLS, which means that our method can achieve better prediction under the same network complexity (i.e., with the same number of the enhancement node). Moreover, we can find that the values of RSD of the proposed CG-DBLS are much higher than that of BLS and FBLS, which implies that our method has better stability than BLS and FBLS. Similar observations can also be found in figures 3-5.

From the above experimental results, we can infer that the proposed CG-DBLS can have a faster convergence rate than the original BLS and its variant FBLS when solving the same problem, and CG-DBLS can achieve a more compact network architecture than other algorithms under the same error level.

Here we give an empirical explanation for the above experimental phenomena. Compared with BLS and FBLS, our method adopts dense connection in the network structure,

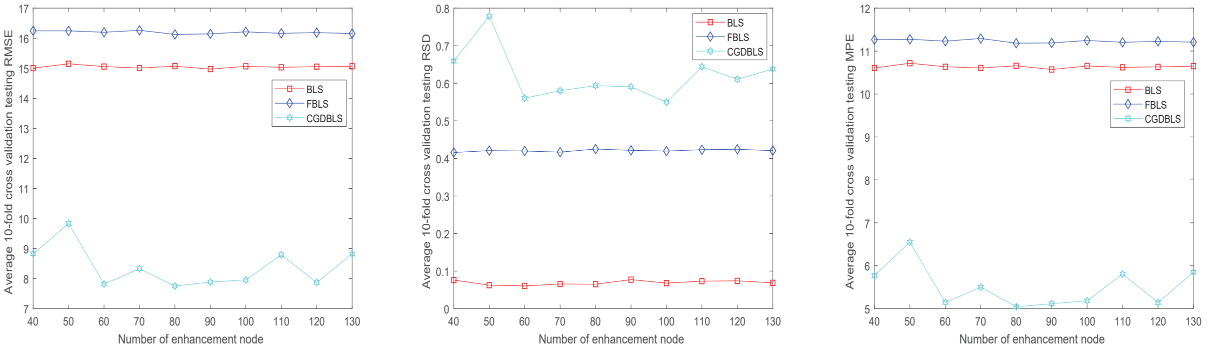

Fig. 2. Average 10-fold cross validation testing RMSE, RSD, and MPE for Airfoil Self-Noise.

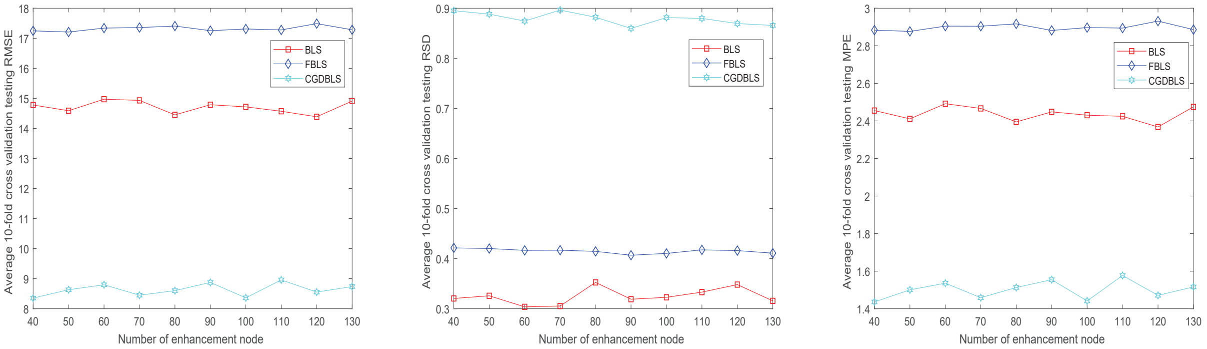

Fig. 3. Average 10-fold cross validation testing RMSE, RSD, and MPE for Combined Cycle Power Plant.

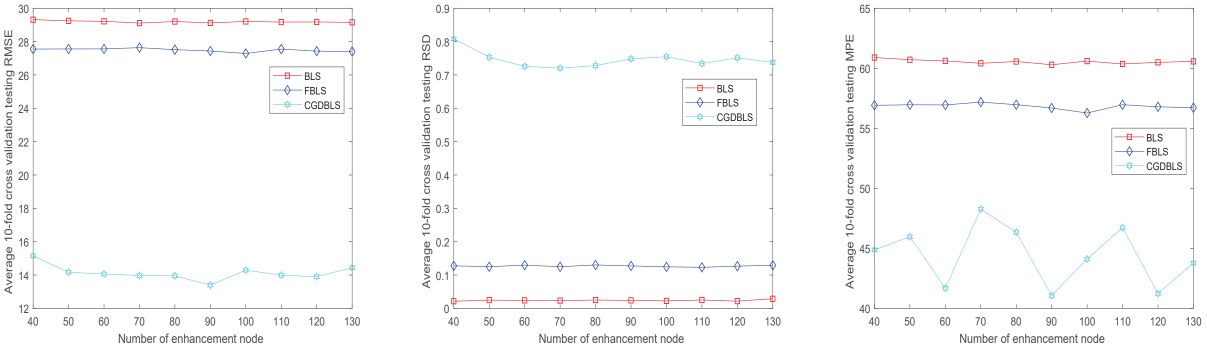

Fig. 4. Average 10-fold cross validation testing RMSE, RSD, and MPE for Concrete Compressive Strength.

which can make the data features get deeper transformation and abstraction; Moreover, we use the recursive least square method to solve the output weights of the feature nodes and use the conjugate gradient method to calculate the output weights of the enhancement nodes, which can make the solution of the model more stable. Therefore, our method can achieve better prediction performance and stability.

V. CONCLUSIONS

In this paper, we proposed an improved BLS algorithm named Dense Broad Learning System based on Conjugate Gradient (CG-DBLS). Different from the original BLS and its variants such as FBLS, in CG-DBLS, the feature nodes and enhancement nodes are directly connected to other the same type of nodes respectively, and the output weights of the feature nodes and enhancement nodes are calculated by using the recursive least square method and the conjugate gradient method respectively. In this way, we can avoid manually selecting the value of λ, thereby improving the stability and generalization ability of the model.

Extensive experimental results on four benchmark data sets show that CG-DBLS can achieve higher accuracy than BLS

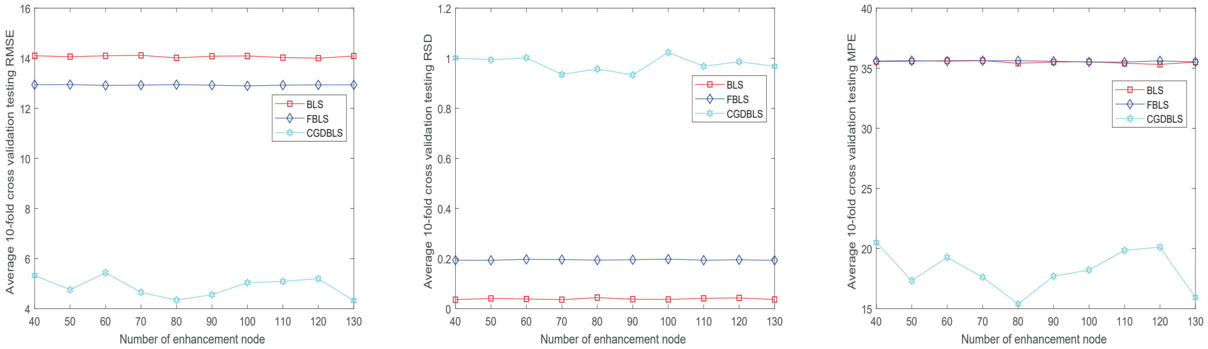

Fig. 5. Average 10-fold cross validation testing RMSE, RSD, and MPE for Energy efficiency.

and FBLS on the same problems, and CG-DBLS has a faster convergence rate and more compact network architecture than other algorithms under the same conditions.

In the future, we would like to give more theoretical proof for CG-DBLS and verify the effectiveness of CG-DBLS on more complex data sets such as image classification tasks and time-series problems [1].

References

[1] W. Cao, Z. Ming, Z. Xu, J. Zhang, and Q. Wang, *Online sequential extreme learning machine with dynamic forgetting factor*, IEEE Access, 7 (2019), pp. 179746–179757.

[2] W. Cao, M. J. A. Patwary, P. Yang, X. Wang, and Z. Ming, *An initial study on the relationship between meta features of dataset and the initialization of nnrw*, in 2019 International Joint Conference on Neural Networks (IJCNN), July 2019, pp. 1–8.

[3] W. Cao, X. Wang, Z. Ming, and J. Gao, *A review on neural networks with random weights*, Neurocomputing, 275 (2018), pp. 278–287.

[4] C. P. Chen and Z. Liu, *Broad learning system: An effective and efficient incremental learning system without the need for deep architecture*, IEEE transactions on neural networks and learning systems, 29 (2017), pp. 10–24.

[5] C. P. Chen, Z. Liu, and S. Feng, *Universal approximation capability of broad learning system and its structural variations*, IEEE transactions on neural networks and learning systems, 30 (2018), pp. 1191–1204.

[6] S. Feng and C. P. Chen, *Broad learning system for control of nonlinear dynamic systems*, in 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2018, pp. 2230–2235.

[7] ———, *Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification*, IEEE transactions on cybernetics, (2018).

[8] R. Fletcher and C. M. Reeves, *Function minimization by conjugate gradients*, The computer journal, 7 (1964), pp. 149–154.

[9] O. Ghahabi and J. Hernando, *Restricted boltzmann machines for vector representation of speech in speaker recognition*, Computer Speech & Language, 47 (2018), pp. 16–29.

[10] J.-W. Jin and C. P. Chen, *Regularized robust broad learning system for uncertain data modeling*, Neurocomputing, 322 (2018), pp. 58–69.

[11] X. Li, J. Shi, X. Dong, and J. Yu, *A new conjugate gradient method based on quasi-newton equation for unconstrained optimization*, Journal of Computational and Applied Mathematics, 350 (2019), pp. 372–379.

[12] Y. Li, W. Zheng, Z. Cui, and T. Zhang, *Face recognition based on recurrent regression neural network*, Neurocomputing, 297 (2018), pp. 50–58.

[13] Z. Liu, B. Chen, B. Xie, H. Qiang, and Z. Zhu, *Feature selection for orthogonal broad learning system based on mutual information*, in 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[14] B. A. Logue and E. Manandhar, *Percent residual accuracy for quantifying goodness-of-fit of linear calibration curves*, Talanta, 189 (2018), pp. 527–533.

[15] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, *Learning and generalization characteristics of the random vector functional-link net*, Neurocomputing, 6 (1994), pp. 163–180.

[16] R. Wang, W. Li, and L. Zhang, *Blur image identification with ensemble convolution neural networks*, Signal Processing, 155 (2019), pp. 73–82.

[17] M. Xu, M. Han, C. P. Chen, and T. Qiu, *Recurrent broad learning systems for time series prediction*, IEEE transactions on cybernetics, (2018).

[18] M. Yan, J. Guo, W. Tian, and Z. Yi, *Symmetric convolutional neural network for mandible segmentation*, Knowledge-Based Systems, 159 (2018), pp. 63–71.

[19] G.-Y. Ye, K.-J. Xu, and W.-K. Wu, *Standard deviation based acoustic emission signal analysis for detecting valve internal leakage*, Sensors and Actuators A: Physical, 283 (2018), pp. 340–347.

[20] T. Zhang, Z. Liu, X.-H. Wang, X.-F. Xing, C. P. Chen, and E. Chen, *Facial expression recognition via broad learning system*, in 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2018, pp. 1898–1902.

[21] Q. Zhou and X. He, *Broad learning model based on enhanced features learning*, IEEE Access, 7 (2019), pp. 42536–42550.