Universal Approximation Capability of Broad Learning System and Its Structural Variations

C. L. Philip Chen, *Fellow, IEEE*, Zhulin Liu, and Shuang Feng

***Abstract*— After a very fast and efficient discriminative broad learning system (BLS) that takes advantage of flatted structure and incremental learning has been developed, here, a mathematical proof of the universal approximation property of BLS is provided. In addition, the framework of several BLS variants with their mathematical modeling is given. The variations include cascade, recurrent, and broad–deep combination structures. From the experimental results, the BLS and its variations outperform several exist learning algorithms on regression performance over function approximation, time series prediction, and face recognition databases. In addition, experiments on the extremely challenging data set, such as MS-Celeb-1M, are given. Compared with other convolutional networks, the effectiveness and efficiency of the variants of BLS are demonstrated.**

***Index Terms*— Broad learning system (BLS), deep learning, face recognition, functional link neural networks (FLNNs), nonlinear function approximation, time-variant big data modeling, universal approximation.**

I. Introduction

WITH the revitalizing of the research in artificial intelligence, recently, many machine learning algorithms that support this development have been introduced viciously. One of the major contributors is the deep learning algorithm, including both generative learning and discriminative learning. One representative of deep generative learning is the restricted Boltzmann machines and its deep model [1]. Another representative discriminative learning is convolutional neural network (CNN) [2]. These deep learning algorithms and models and their variants have carved out a research wave in machine learning for people to follow. The learning algorithms also have been used in pattern recognition, image recognition, speech recognition, and video processing applications and exhibited outstanding performance.

The CNN structure is a kind of multilayer neural networks with convolution and pooling operations as feature mapping that can be considered to have the similar feature extraction property with properly adjusted weights. One may also consider that the feature mapping portion is a kind of kernel mappings, in which different kernels may be used to replace convolution and pooling operators. CNN and its variants have successfully accomplished a very high recognition rate in many recognition competitions tested on ImageNet data set.

Although the deep structure has been so powerful, most networks suffer from time-consuming training process because complicated structures are involved. Many of the studies require high-performance computing and powerful facility. Moreover, this complication makes it so difficult to analyze the deep structure theoretically that most work spans in turning the parameters or stacking more layers for better accuracy.

Recently, a very fast and efficient discriminative learning—broad learning system (BLS)—has been developed by Chen and Liu [3]. Without stacking the layer-structure, the designed neural networks expand the neural nodes broadly and update the weights of the neural networks incrementally when additional nodes are needed and when the input data entering to the neural networks continuously. Therefore, the BLS structure is perfectly suitable for modeling and learning in a time-variant big data environment. It is also indicated that BLS significantly outperforms existing deep structures in learning accuracy and generalization ability in Modified National Institute of Standards and Technology (MNIST) and handwriting recognition and New York University object recognition benchmark (NORB) database [4].

In addition to BLS fine discriminative capability in classification and recognition, here, a mathematical proof of the universal approximation property of BLS is provided. Based on the theorem, it is stated that BLS is a nonlinear function approximator. With this, the regression performance of BLS is compared with support vector machine (SVM), least squared SVM (LSSVM), and extreme learning machine (ELM) on several benchmarked data sets on function approximation, time series prediction, and face recognition.

This paper also discusses several BLS variants, where different weight connections are established within the feature mapping nodes, within enhancement nodes and between feature nodes and enhancement nodes. Mathematical modeling of these variants is also given. It is hoped that these variant architectures can be used for future research.

In the following, the preliminary knowledge for this paper is introduced followed by the proof of the universal approximation property of the BLS and followed by variant

Manuscript received October 25, 2017; revised May 14, 2018 and July 13, 2018; accepted August 15, 2018. Date of publication September 10, 2018; date of current version March 18, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61751202, Grant 61751205, and Grant 61572540, in part by the Macau Science and Technology Development Fund under Grant 019/2015/A1, Grant 079/2017/A2, and Grant 024/2015/AMJ, and in part by the University of Macau through the Multiyear Research Grants. *(Corresponding author: Zhulin Liu.)*

C. L. P. Chen is with the Faculty of Science and Technology, University of Macau, Macau 99999, China, also with the College of Navigation, Dalian Maritime University, Dalian 116026, China, and also with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China (e-mail: philip.chen@ieee.org).

Z. Liu is with the Faculty of Science and Technology, University of Macau, Macau 99999, China (e-mail: zhulinlau@gmail.com).

S. Feng is with the School of Applied mathematics, Beijing Normal University, Zhuhai 519087, China, and also with the Faculty of Science and Technology, University of Macau, Macau 99999, China (e-mail: fengshuang@bnuz.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2866622

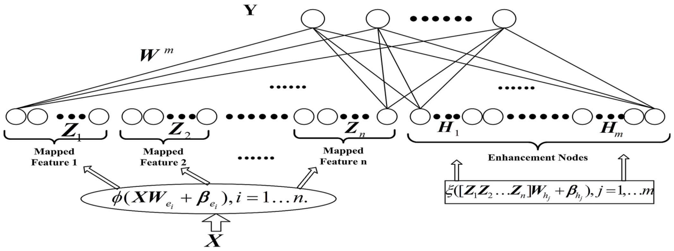


Fig. 1. Framework of a typical BLS.

BLS architectures and experiments in function approximation, time series prediction, and face recognition benchmarks. To demonstrate the advantages of variants of BLS, experiments on the typically large-scale data sets are followed. Among them, the most challenging one is the MS-Celeb-1M data set. The results prove that the cascade convolution feature mapping nodes BLS outperforms other convolutional networks.

II. Preliminaries

A. Functional Link Neural Networks

The functional link neural network (FLNN) proposed by Klassen *et al.* [5] is a variant of the higher order neural network without hidden units, and it has been further developed to the random vector functional link network [6]. Different various improvements and models as well as successful applications of FLNN are developed due to its universal approximation properties (see [7]–[9]). A comprehensive review of FLNN can be referred to [10].

Chen [11] and Chen *et al.* [12] have presented an adaptive implementation of the FLNN architecture together with a supervised learning algorithm named rank-expansion with instant learning. The advantage of this rapid algorithm is that it can learn the weights in a one-shot training phase without iteratively updating the parameters. In addition, a fast learning algorithm is proposed in [13] to find optimal weights of the flat neural networks (especially, the functional link network), which adopts the linear least-square method, and this algorithm makes it easier to update the weights instantly for incremental input patterns and enhancement nodes.

B. Broad Learning Systems

The BLS [3] provides an alternative way of learning deep structures that usually suffer from time-consuming training of abundant parameters in the filters and layers. The training process is through incremental learning algorithms for fast remodeling in broad expansion without a retraining process if the network deems to be expanded.

The BLS depicted in Fig. 1 is established in the form of a flat network, where the original inputs are transformed into random features in "feature nodes" and the structure is expanded in the wide sense in the "enhancement nodes." Specifically, the main character is that in a BLS, the input data are first transformed into random features by some feature mappings that are further connected by nonlinear activation functions to form the enhancement nodes. Then, the random features (nodes) together with the outputs of the enhancement layer are connected to the output layer, where this output layer weights are to be determined by either a fast pseudoinverse of the system equation or an iterative gradient descent training algorithm. Incremental learning algorithms are used for a new input arriving or an expansion of the enhancement nodes. This characteristic makes the BLS very efficient and much less time-consuming compared with multilayer perceptron and deep structures, such as CNN, deep belief networks, deep Boltzmann machines, stacked auto encoders, and stacked deep auto encoders.

We will briefly describe the establishment of a typical BLS, and readers can refer to [3] for details. Given the training data $\{\boldsymbol{X}, \hat{\boldsymbol{Y}}\} \in \mathbb{R}^{N\times(M+C)}$ and n feature mappings $\phi_i, i = 1, 2, \ldots, n$, then the ith mapped features are

$$\boldsymbol{Z}_i = \phi_i(\boldsymbol{X}\boldsymbol{W}_{e_i} + \boldsymbol{\beta}_{e_i}), \quad i = 1, 2, \ldots, n \tag{1}$$

where the weights $\boldsymbol{W}_{e_i}$ and bias term $\boldsymbol{\beta}_{e_i}$ are randomly generated matrices with the proper dimensions.

We denote $\boldsymbol{Z}^n \triangleq [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n]$ as the collection of n groups of feature nodes. Then, $\boldsymbol{Z}^n$ is connected into the layer of enhancement nodes.

Similarly, we denote the outputs of the jth group of enhancement nodes by

$$\boldsymbol{H}_j \triangleq \xi_j(\boldsymbol{Z}^n\boldsymbol{W}_{h_j} + \boldsymbol{\beta}_{h_j}), \quad j = 1, 2, \ldots, m \tag{2}$$

where ξ_j is a nonlinear activation function. In addition, we denote the outputs of the enhancement layer by $\boldsymbol{H}^m \triangleq [\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m]$.

For simplicity and without loss of generality, we will omit the subscripts of the feature mapping ϕ_i and the activation function ξ_j in the following part. However, ϕ_i can be selected differently in establishing a model as well as ξ_j.

In order to obtain sparse representation of input data, the randomly initialized weight matrix $\boldsymbol{W}_{e_i}$ is fine-tuned by applying the linear inverse problem (please refer to [3, eq. (4)]).

Therefore, the output $\boldsymbol{Y}$ of a BLS has the following form:

$$\begin{aligned}\boldsymbol{Y} &= [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n, \boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m]\boldsymbol{W}^m \\ &= [\boldsymbol{Z}^n, \boldsymbol{H}^m]\boldsymbol{W}^m\end{aligned} \tag{3}$$

where $\boldsymbol{W}^m$ are the weights connecting the layer of feature nodes and the layer of enhancement nodes to the output layer, and $\boldsymbol{W}^m \triangleq [\boldsymbol{Z}^n, \boldsymbol{H}^m]^+\boldsymbol{Y}$. Here, $\boldsymbol{W}^m$ can be easily computed using the ridge regression approximation of pseudoinverse $[\boldsymbol{Z}^n, \boldsymbol{H}^m]^+$.

C. Incremental Learning Algorithms for Broad Learning Systems

Three incremental learning algorithms are also developed for the BLS without retraining the whole model [3],

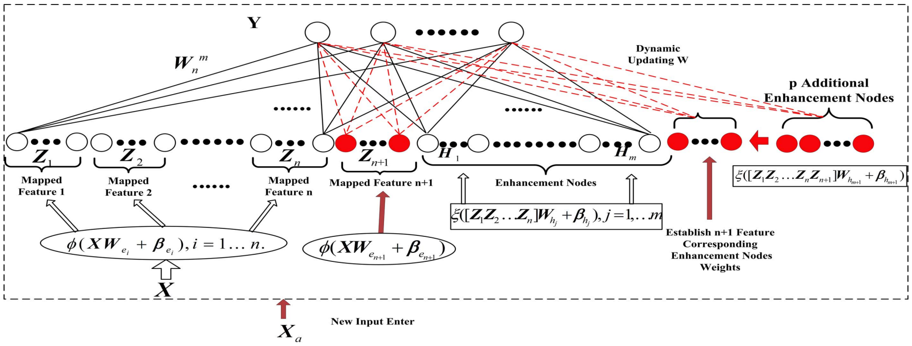

Y
W_n^m
Dynamic Updating W
p Additional Enhancement Nodes
Z_1
Z_2
Z_n
Z_{n+1}
H_1
H_m
Mapped Feature 1
Mapped Feature 2
Mapped Feature n
Mapped Feature n+1
Enhancement Nodes
$\xi([Z_1Z_2\ldots Z_nZ_{n+1}]W_{h_{m+1}}+\beta_{h_{m+1}})$
$\xi([Z_1Z_2\ldots Z_n]W_{h_j}+\beta_{h_j}), j=1,\ldots m$
$\phi(XW_{e_i}+\beta_{e_i}), i=1\ldots n.$
$\phi(XW_{e_{n+1}}+\beta_{e_{n+1}})$
X
Establish n+1 Feature Corresponding Enhancement Nodes Weights
X_a
New Input Enter

Fig. 2. BLS with increment of input variables, feature nodes, and enhancement nodes (red parts).

which deals with three scenarios, including the increments of enhancement nodes, feature nodes, and input data (see Fig. 2). We will give a brief description of the three incremental conditions as follows.

1) Increment of Enhancement Nodes: Suppose that we expand the BLS by adding p enhancement nodes, and denote $\boldsymbol{A}^m \triangleq [\boldsymbol{Z}^n, \boldsymbol{H}^m]$ and

$$\boldsymbol{A}^{m+1} \triangleq [\boldsymbol{A}^m, \xi(\boldsymbol{Z}^n \boldsymbol{W}_{h_{m+1}} + \boldsymbol{\beta}_{h_{m+1}})] \tag{4}$$

where $\boldsymbol{W}_{h_{m+1}}$ and $\boldsymbol{\beta}_{h_{m+1}}$ are randomly generated weights and bias terms connecting feature nodes to the p additional enhancement nodes.

Then, the new weights of this incremental BLS can be calculated by

$$\boldsymbol{W}^{m+1} \triangleq (\boldsymbol{A}^{m+1})^+ \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{W}^m - \boldsymbol{D}\boldsymbol{B}^T\boldsymbol{Y} \\ \boldsymbol{B}^T\boldsymbol{Y} \end{bmatrix} \tag{5}$$

where the pseudoinverse of the new matrix $\boldsymbol{A}^{m+1}$ is

$$(\boldsymbol{A}^{m+1})^+ = \begin{bmatrix} (\boldsymbol{A}^m)^+ - \boldsymbol{D}\boldsymbol{B}^T \\ \boldsymbol{B}^T \end{bmatrix} \tag{6}$$

and

$$\begin{aligned} \boldsymbol{B}^T &= \begin{cases} \boldsymbol{C}^+, & \text{if } \boldsymbol{C} \neq 0 \\ (\boldsymbol{1} + \boldsymbol{D}^T\boldsymbol{D})^{-1}\boldsymbol{D}^T(\boldsymbol{A}^m)^+, & \text{if } \boldsymbol{C} = 0 \end{cases} \\ \boldsymbol{C} &= \xi(\boldsymbol{Z}^n\boldsymbol{W}_{h_{m+1}} + \boldsymbol{\beta}_{h_{m+1}}) - \boldsymbol{A}^m\boldsymbol{D} \\ \boldsymbol{D} &= (\boldsymbol{A}^m)^+\xi(\boldsymbol{Z}^n\boldsymbol{W}_{h_{m+1}} + \boldsymbol{\beta}_{h_{m+1}}). \end{aligned} \tag{7}$$

We can see from the above-mentioned formulae that it only needs to compute the pseudoinverse of necessary components instead of calculating that of the whole $\boldsymbol{A}^{m+1}$, which generates the fast learning property of BLS.

2) Increment of Feature Nodes: Sometimes a discriminative model (shadow or deep) may suffer from insufficient features that cannot well represent the input data. A traditional solution for these architectures is to extract more new features by increasing the number of filters or layers and train the models from the very beginning, which is very time-consuming, especially, for some deep models.

However, it is convenient to implement the increment of a new feature mapping in BLS. The new few mapping nodes can be easily inserted into the structure and the connection weights can be easily trained by the incremental learning algorithm similar to that of adding new above-mentioned enhancement nodes.

Assume that the initial BLS consists of n groups of feature nodes and m groups of enhancement nodes, respectively, now, we consider that the $(n+1)$th group of feature nodes are added and denoted by

$$\boldsymbol{Z}_{n+1} = \phi(\boldsymbol{X}\boldsymbol{W}_{e_{n+1}} + \boldsymbol{\beta}_{e_{n+1}}). \tag{8}$$

In addition, the output of corresponding enhancement nodes is

$$\boldsymbol{H}_{\mathrm{ex}_m} \triangleq [\xi(\boldsymbol{Z}_{n+1}\boldsymbol{W}_{\mathrm{ex}_1} + \boldsymbol{\beta}_{\mathrm{ex}_1}), \ldots, \xi(\boldsymbol{Z}_{n+1}\boldsymbol{W}_{\mathrm{ex}_m} + \boldsymbol{\beta}_{\mathrm{ex}_m})] \tag{9}$$

where $\boldsymbol{W}_{\mathrm{ex}_i}$ and $\boldsymbol{\beta}_{\mathrm{ex}_i}$ are randomly generated weights and bias terms that connect the newly added feature nodes to the enhancement nodes.

Now, let $\boldsymbol{A}_{n+1}^m \triangleq [\boldsymbol{A}^m, \boldsymbol{Z}_{n+1}, \boldsymbol{H}_{\mathrm{ex}_m}]$, and we only have to compute the pseudoinverse of a matrix containing $[\boldsymbol{Z}_{n+1}, \boldsymbol{H}_{\mathrm{ex}_m}]$ to obtain the new weights $\boldsymbol{W}_{n+1}^m$ of this incremental BLS. The formulae for $\boldsymbol{W}_{n+1}^m$ are similar to (5), so we omit them here.

3) Increment of Input Data: In some online learning scenarios, the training data are keep coming into the system and we should establish a model that is adaptive to the new data. A common way for deep models is to retrain them again with the whole training data. On the contrary, the BLS can be easily adapted to the new training data by only updating the corresponding part of weights for the newly added input samples. See details in the following.

Suppose that $\{\boldsymbol{X_a}, \boldsymbol{Y_a}\}$ denotes the new training data added to a BLS. The generated feature nodes for $\boldsymbol{X_a}$ are

$$\boldsymbol{Z}_a^n = [\phi(\boldsymbol{X_a}\boldsymbol{W}_{e_1} + \boldsymbol{\beta}_{e_1}), \ldots, \phi(\boldsymbol{X_a}\boldsymbol{W}_{e_n} + \boldsymbol{\beta}_{e_n})] \tag{10}$$

and the output matrix of feature and enhancement layers for the new data are denoted as

$$\boldsymbol{A_x} \triangleq \left[\boldsymbol{Z}_a^n, \xi\left(\boldsymbol{Z}_x^n\boldsymbol{W}_{h_1} + \boldsymbol{\beta}_{h_1}\right), \ldots, \xi\left(\boldsymbol{Z}_x^n\boldsymbol{W}_{h_m} + \boldsymbol{\beta}_{h_m}\right)\right]. \tag{11}$$

The weights of this incremental BLS can be updated by

$$\boldsymbol{W}_a^m = \boldsymbol{W}^m + \left(\boldsymbol{Y}_{\boldsymbol{a}}^T - \boldsymbol{A}_{\boldsymbol{x}}^T\boldsymbol{W}^m\right)\boldsymbol{B} \tag{12}$$

where

$$\begin{aligned} \boldsymbol{B}^T &= \begin{cases} \boldsymbol{C}^+, & \text{if } \boldsymbol{C} \neq 0 \\ (\boldsymbol{1} + \boldsymbol{D}^T\boldsymbol{D})^{-1}(\boldsymbol{A}^m)^+\boldsymbol{D}, & \text{if } \boldsymbol{C} = 0 \end{cases} \\ \boldsymbol{C} &= \boldsymbol{A}_{\boldsymbol{x}}^T - \boldsymbol{D}^T\boldsymbol{A}^m \\ \boldsymbol{D}^T &= \boldsymbol{A}_{\boldsymbol{x}}^T(\boldsymbol{A}^m)^+. \end{aligned} \tag{13}$$

Similarly, this incremental training process saves time since it only computes the pseudoinverse of matrix containing the new part $\boldsymbol{A_x}$. This particular scheme is suitable and effective for system modeling that requires online learning.

III. Universal Approximation Property of BLS

We have demonstrated the fine discriminative capability of BLS in [3] through some representative benchmarks for classification. We will discuss the universal approximation property of BLS and prove several theorems in this section.

Similar to the denotations in [3, Th. 1], consider any continuous function $f \in C(\mathbf{I}^d)$, which defined on the standard hypercube $\mathbf{I}^d = [0;1]^d \subset \mathbb{R}^d$, the BLS with nonconstant bounded feature mapping ϕ and activation function ξ can equivalently be denoted as

$$\begin{aligned} f_{\boldsymbol{w}_{m,n}}(\boldsymbol{x}) &= \sum_{i=1}^{n*k} w_i \phi(\boldsymbol{x}\boldsymbol{w}_{e_i} + \beta_{e_i}) \\ &\quad + \sum_{j=1}^{m*q} w_{nk+j}\xi(\boldsymbol{z}\boldsymbol{w}_{h_j} + \beta_{h_j}) \\ &= \sum_{i=1}^{n*k} w_i \phi(\boldsymbol{x}\boldsymbol{w}_{e_i} + \beta_{e_i}) \\ &\quad + \sum_{j=1}^{m*q} w_{nk+j}\xi(\boldsymbol{x}; \{\phi, \boldsymbol{w}_{h_j}, \beta_{h_j}\}) \end{aligned}$$

where $\boldsymbol{z} = [\phi(\boldsymbol{x}\boldsymbol{w}_{e_1} + \beta_{e_1}), \ldots, \phi(\boldsymbol{x}\boldsymbol{w}_{e_{nk}} + \beta_{e_{nk}})]$, and

$$\begin{aligned} \boldsymbol{w}_{m,n} = (&n, m, w_1, \ldots, w_{nk+mq}, \boldsymbol{w}_{e_1}, \ldots, \boldsymbol{w}_{e_{nk}}, \boldsymbol{w}_{h_1}, \\ &\ldots, \boldsymbol{w}_{h_{mq}}, \beta_{e_1}, \ldots, \beta_{e_{nk}}, \beta_{h_1}, \ldots, \beta_{h_{mq}}) \end{aligned}$$

is the set of overall parameters for the functional link network. Among them, the randomly generated part is denoted as $\boldsymbol{\lambda}_{m,n} = (\boldsymbol{w}_{e_1}, \ldots, \boldsymbol{w}_{e_{nk}}, \boldsymbol{w}_{h_1}, \ldots, \boldsymbol{w}_{h_{mq}}, \beta_{e_1}, \ldots, \beta_{e_{nk}}, \beta_{h_1}, \ldots, \beta_{h_{mq}})$. Assume that the random variables are defined on the probability measure $\mu_{m,n}$, and notation E is the expectation with respect to the probability measure. Moreover, the distance between the approximation function $f_{\boldsymbol{w}_{m,n}}$ and the function f on the compact set $\mathbf{K} \subset \mathbf{I}^d$ can be denoted as

$$\rho_{\mathbf{K}}(f, f_{\boldsymbol{w}_{m,n}}) = \sqrt{E\left[\int_{\mathbf{K}} (f(\boldsymbol{x}) - f_{\boldsymbol{w}_{m,n}}(\boldsymbol{x}))^2 d\boldsymbol{x}\right]}.$$

Our main result is as follows.

Theorem 1: For any compact set $\mathbf{K} \subset \mathbf{I}^d$ and any continuous function f in $C(\mathbf{I}^d)$, there exists a sequence of $\{f_{\boldsymbol{w}_{m,n}}\}$ in BLS that is constructed by nonconstant bounded feature mapping ϕ and absolutely integrable activation function ξ (functions on $\mathbf{I}^d$, such that $\int_{\mathbb{R}^d} \xi^2(\boldsymbol{x})d\boldsymbol{x} < \infty$) and a respective sequence of probability measures $\mu_{m,n}$, such that

$$\lim_{m,n\to\infty} \rho_{\mathbf{K}}(f, f_{\boldsymbol{w}_{m,n}}) = 0.$$

Moreover, the randomly generated parameters $\boldsymbol{\lambda}_{m,n}$ are samples from the distributions of probability measures $\mu_{m,n}$.

Proof: Recall that for function f, the approximation solution of BLS is the function $f_{\boldsymbol{w}_{m,n}}$ defined earlier. Let $w_z = [w_{z_1}, \ldots, w_{z_{nk}}]$ denote the weight matrix connecting the feature nodes $\boldsymbol{Z}^n$ to the output layer, and let $w_{\boldsymbol{h}} = [w_{\boldsymbol{h}_1}, \ldots, w_{\boldsymbol{h}_{mq}}]$ denote the weight matrix connecting the enhancement nodes $\boldsymbol{H}^m$ to the output layer.

Therefore, for any integer n, define

$$f_{\boldsymbol{w}_z} = \sum_{i=1}^{nk} w_{z_i}\phi(\boldsymbol{x}\boldsymbol{w}_{e_i} + \beta_{e_i})$$

where $\boldsymbol{w}_{e_i}, \beta_{e_i}, i = 1, \ldots, nk$, are samples from the given probability measures. Obviously, the resident function $f_{r_n} = f - f_{\boldsymbol{w}_z}$ is bounded and integrable in $\mathbf{I}^d$ since the feature mapping ϕ is bounded. Furthermore, there exists a function $f_{c_n} \in C(\mathbf{I}^d)$, such that $\forall\varepsilon > 0$, we have

$$\rho_{\mathbf{K}}(f_{c_n}, f_{r_n}) < \frac{\varepsilon}{2}.$$

The above-mentioned conclusion could be theoretically guaranteed by the fact in [14]. It is clear that for any $f_{c_n} \in L^2(\boldsymbol{K})$, there exists a smooth function f_{r_n}, such that $||f_{c_n} - f_{r_n}|| < \epsilon'$.

Hence, to approximate the resident function f_{c_n}, define that

$$f_{\boldsymbol{w_h}} = \sum_{j=1}^{mq} w_{\boldsymbol{h}_j}\xi(\boldsymbol{x}; \{\phi, \boldsymbol{w}_{h_j}, \beta_{h_j}\})$$

where $\boldsymbol{w}_{h_j}, \beta_{h_j}, j = 1, \ldots, mq$, are samples from the given probability measures. Since ϕ and ξ are nonconstant and bounded, the composition function $\xi(\boldsymbol{x}; \{\phi, \boldsymbol{w}_{h_j}, \beta_{h_j}\}), j = 1, \ldots, mq$ is obviously absolutely integrable. Hence, according to the universal approximation property of RVFL (details please refer to [9, Th. 1]), there exists a sequence of $f_{\boldsymbol{w_h}}$, such that $\forall\varepsilon > 0$, we have

$$\rho_{\mathbf{K}}(f_{c_n}, f_{\boldsymbol{w_h}}) < \frac{\varepsilon}{2}.$$

Finally, we have that

$$\begin{aligned} \rho_{\mathbf{K}}(f, f_{\boldsymbol{w}_{m,n}}) &= \sqrt{E\left[\int_{\mathbf{K}} (f(\boldsymbol{x}) - f_{\boldsymbol{w}_{m,n}}(\boldsymbol{x}))^2 d\boldsymbol{x}\right]} \\ &= \sqrt{E\left[\int_{\mathbf{K}} ((f(\boldsymbol{x}) - f_{\boldsymbol{w}_z}(\boldsymbol{x})) - f_{\boldsymbol{w_h}}(\boldsymbol{x}))^2 d\boldsymbol{x}\right]} \\ &= \rho_{\mathbf{K}}(f_{r_n}, f_{\boldsymbol{w_h}}) \\ &\le \rho_{\mathbf{K}}(f_{r_n}, f_{c_n}) + \rho_{\mathbf{K}}(f_{c_n}, f_{\boldsymbol{w_h}}) \\ &\le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} \\ &= \varepsilon. \end{aligned}$$

Hence, we could conclude that

$$\lim_{m,n\to\infty} \rho_{\mathbf{K}}(f, f_{\boldsymbol{w}_{m,n}}) = 0.\square$$

Corollary 1: For any compact set $\mathbf{K} \subset \mathbf{I}^d$ and any measurable function f in $\mathbf{I}^d$, there exists a sequence of $\{f_{\boldsymbol{w}_{m,n}}\}$ in BLS that is constructed by nonconstant bounded feature mapping ϕ and absolutely integrable activation function ξ (functions on $\mathbf{I}^d$, such that $\int_{\mathbb{R}^d} \xi^2(\boldsymbol{x})d\boldsymbol{x} < \infty$) and a respective sequence of probability measures $\mu_{m,n}$, such that

$$\lim_{m,n\to\infty} \rho_{\mathbf{K}}(f, f_{\boldsymbol{w}_{m,n}}) = 0.$$

Moreover, the randomly generated parameters $\boldsymbol{\lambda}_{m,n}$ are samples from the distributions of probability measures $\mu_{m,n}$.

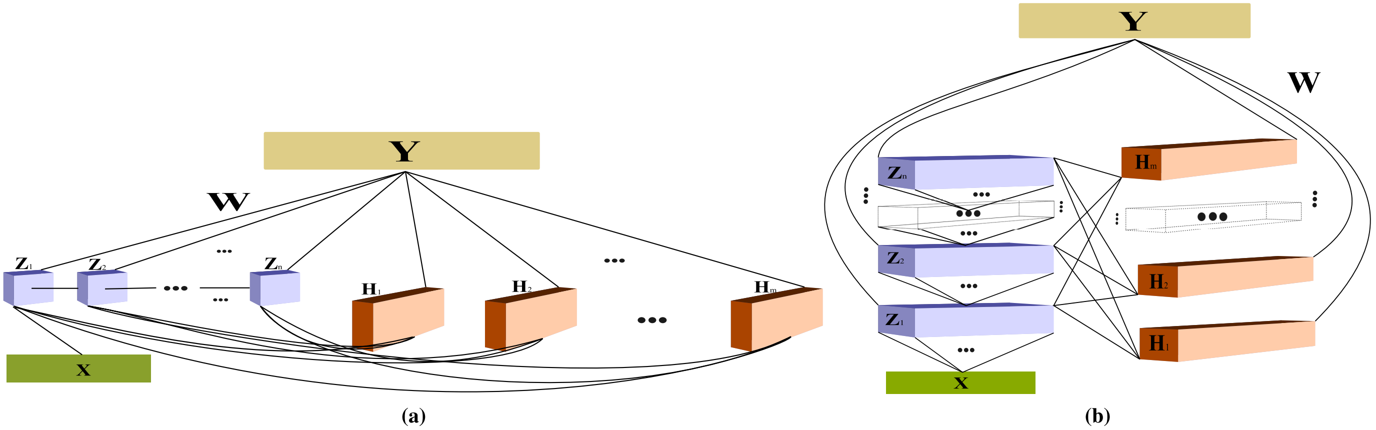


Fig. 3. CFBLS. Cascade of feature mapping nodes. (a) Broad structure. (b) Structure redrawn, as left side showing the cascade architecture.

This corollary obviously holds since it is clear that for any $f \in L^2(\boldsymbol{K})$, there exists a continuous function g, such that $||f - g|| < \epsilon$ [14].

The above-mentioned Theorem states that the BLS is a nonlinear function approximator but the weights are to be found yet sometimes may not be easy. In the following, we will discuss different composite models of BLS. These composite models establish a different connections on either feature mapping nodes or enhancement nodes that create more nonlinearity mapping for the input and therefore may be easier to find the connection weight in the last layer.

IV. Composite Models of Broad Learning System

The BLS is a flexible model to be modified under various constraints. Among them, regularization is of great benefit to specific applications (see [15], [16]). A variant, which is named graph regularized BLS for image recognition, has been proposed in [17] and [18]. Additional structures based on the original BLS would be proposed in this section. Several models are illustrated and discussed. As usual, the subscriptions of the adopted functions $\phi(\cdot)$ for the construction of the feature nodes, and the functions $\xi(\cdot)$ for the enhancement nodes are omitted. Generally, the variants of the model are motivated by the following consideration: 1) the cascade of the feature maps (CFBLS); 2) the cascade of the enhancement nodes (CEBLS); 3) the limited connection between the groups of cascaded feature maps and the enhancement nodes (LCFBLS); 4) the limited connection between the feature maps and the groups of cascaded enhancement nodes (LCEBLS); and (5) the cascade of feature mapping nodes and enhancement nodes (CFEBLS).

A. Broad Learning Systems: Cascade of Feature Mapping Nodes (CFBLS)

This architecture cascades the a group of feature mapping nodes one after another. As seen in Fig. 3(a), the feature mapping nodes $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \dots, \boldsymbol{Z}_n$ form a cascade connections.

Therefore, for the input data $\boldsymbol{X}$, the first group of feature mapping nodes $\boldsymbol{Z}_1$ is denoted as

$$\boldsymbol{Z}_1 = \phi(\boldsymbol{X}\boldsymbol{W}_{e_1} + \boldsymbol{\beta}_{e_1}) \triangleq \phi(\boldsymbol{X}; \{\boldsymbol{W}_{e_1}, \boldsymbol{\beta}_{e_1}\})$$

where $\boldsymbol{W}_{e_1}$ and $\boldsymbol{\beta}_{e_1}$ are randomly generated by distribution $\rho(w)$. As for the second group, the feature mapping nodes $\boldsymbol{Z}_2$ are established using the output of the $\boldsymbol{Z}_1$ nodes; therefore, $\boldsymbol{Z}_2$ is expressed as

$$\begin{aligned}\boldsymbol{Z}_2 &= \phi(\boldsymbol{Z}_1\boldsymbol{W}_{e_2} + \boldsymbol{\beta}_{e_2})\\ &= \phi(\phi(\boldsymbol{X}\boldsymbol{W}_{e_1} + \boldsymbol{\beta}_{e_1})\boldsymbol{W}_{e_2} + \boldsymbol{\beta}_{e_2})\\ &\triangleq \phi^2(\boldsymbol{X}; \{\boldsymbol{W}_{e_i}, \boldsymbol{\beta}_{e_i}\}_{i=1,2}).\end{aligned}$$

Using the same process continuously, all the n groups of feature mapping nodes are formulated as

$$\begin{aligned}\boldsymbol{Z}_k &= \phi(\boldsymbol{Z}_{k-1}\boldsymbol{W}_{e_k} + \boldsymbol{\beta}_{e_k})\\ &\triangleq \phi^k(\boldsymbol{X}; \{\boldsymbol{W}_{e_i}, \boldsymbol{\beta}_{e_i}\}_{i=1}^{k}), \quad \text{for } k = 1, \dots, n \qquad (14)\end{aligned}$$

where $\boldsymbol{W}_{e_i}$ and $\boldsymbol{\beta}_{e_i}$ are randomly generated.

Next, the concentrated feature nodes $\boldsymbol{Z}^n \triangleq [\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_n]$ are connected with the enhancement nodes $\{H_j\}_{j=1}^m$, where

$$\boldsymbol{H}_j \triangleq \xi(\boldsymbol{Z}^n\boldsymbol{W}_{h_j} + \boldsymbol{\beta}_{h_j})$$

and $\boldsymbol{W}_{h_j}$ and $\boldsymbol{\beta}_{h_j}$ are under the distribution $\rho_e(w)$. Here, the distributions $\rho_e(w)$ and $\rho(w)$ usually are usually equal.

Finally, suppose that the network consists of n groups of feature nodes and m groups of enhancement nodes, the system model of this cascade of feature nodes BLS is summarized as follows:

$$\begin{aligned}\boldsymbol{Y} &= [\phi(\boldsymbol{X}; \{\boldsymbol{W}_{e_1}, \boldsymbol{\beta}_{e_1}\}), \dots, \phi^n\big(\boldsymbol{X}; \{\boldsymbol{W}_{e_i}, \boldsymbol{\beta}_{e_i}\}_{i=1}^n\big)\\ &\quad |\xi(\boldsymbol{Z}^n\boldsymbol{W}_{h_1} + \boldsymbol{\beta}_{h_1}), \quad \dots, \xi(\boldsymbol{Z}^n\boldsymbol{W}_{h_m} + \boldsymbol{\beta}_{h_m})]\boldsymbol{W}_n^m\\ &= [\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_n|\boldsymbol{H}_1, \dots, \boldsymbol{H}_m]\boldsymbol{W}_n^m\\ &= [\boldsymbol{Z}^n|\boldsymbol{H}^m]\boldsymbol{W}_n^m\end{aligned}$$

where $\boldsymbol{H}^m \triangleq [\boldsymbol{H}_1, \dots, \boldsymbol{H}_m]$, and $\boldsymbol{W}_n^m$ is calculated through the pseudoinverse of $[\boldsymbol{Z}^n|\boldsymbol{H}^m]$.

The incremental model of this composite network can be derived similarly and is described in the following.

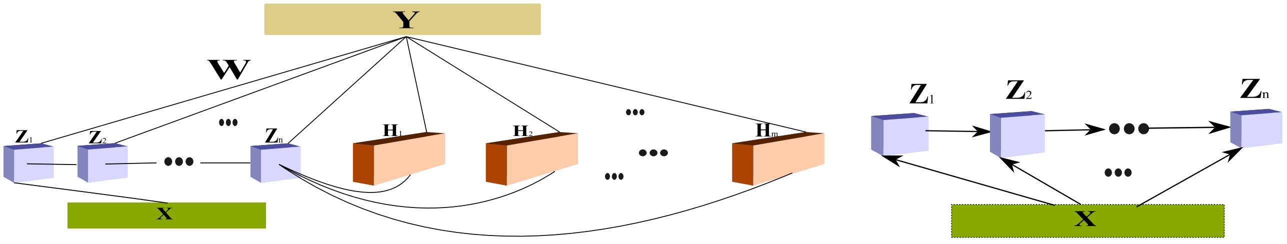
First, if the $(n+1)$th set of composite feature nodes is incrementally added and denoted as

$$\boldsymbol{Z}_{n+1} \triangleq \phi^{n+1}\big(\boldsymbol{X}; \{\boldsymbol{W}_{e_i}, \boldsymbol{\beta}_{e_i}\}_{i=1}^{n+1}\big).$$

Consequently, the m groups of enhancement nodes are updated under the randomly generated weights

$$\boldsymbol{H}_{\mathrm{ex}_m} \triangleq [\xi(\boldsymbol{Z}_{n+1}\boldsymbol{W}_{\mathrm{ex}_1} + \boldsymbol{\beta}_{\mathrm{ex}_1}), \dots, \xi(\boldsymbol{Z}_{n+1}\boldsymbol{W}_{\mathrm{ex}_m} + \boldsymbol{\beta}_{\mathrm{ex}_m})]$$

where $\boldsymbol{W}_{\mathrm{ex}_i}, \boldsymbol{\beta}_{\mathrm{ex}_i}, i = 1, \dots, m$ are randomly generated.



(a) (b)

Fig. 4. LCFBLS. Cascade of feature mapping nodes with the last group connected to the enhancement nodes. (a) Broad structure. (b) Alternative feature nodes connection (recurrent structure).

Second, if the $(m+1)$th group of the enhancement nodes are incrementally added to the system and are denoted as

$$H_{m+1} \triangleq [\xi(Z^{n+1}W_{h_{m+1}} + \beta_{h_{m+1}})]$$

where $Z^{n+1} \triangleq [Z_1, \ldots, Z_{n+1}]$, and $W_{h_{m+1}}, \beta_{h_{m+1}}$ are randomly generated. Denote $A_n^m \triangleq [Z^n|H^m]$ and $A_{n+1}^{m+1} \triangleq [A_n^m|Z_{n+1}|H_{\mathrm{ex}_m}|H_{m+1}]$, the updated pseudoinverse and the new weights of this cascade BLS network should be

$$\left(A_{n+1}^{m+1}\right)^+ = \begin{bmatrix} \left(A_n^m\right)^+ - DB^T \\ B^T \end{bmatrix} \tag{15}$$

$$W_{n+1}^{m+1} = \begin{bmatrix} W_n^m - DB^TY \\ B^TY \end{bmatrix} \tag{16}$$

where $D = (A_n^m)^+[Z_{n+1}|H_{\mathrm{ex}_m}|H_{m+1}]$

$$B^T = \begin{cases} (C)^+, & \text{if } C \neq 0 \\ (1 + D^TD)^{-1}D^T\left(A_n^m\right)^+, & \text{if } C = 0 \end{cases} \tag{17}$$

and $C = [Z_{n+1}|H_{\mathrm{ex}_m}|H_{m+1}] - A_n^m D$.

Specifically, this network inherits the advantage of fast incremental learning in BLS. Besides, more characteristic features are built in the modified network that leads to a more versatile system. Fig. 3(a) is the illustration of the above-mentioned broad learning network, and Fig. 3(b) is the equivalent illustration of this cascade BLS network.

B. Broad Learning Systems: Cascade of Feature Mapping Nodes With Its Last Group Connects to the Enhancement Nodes (LCFBLS) or Recurrent Feature Nodes

Section IV-A describes a modified cascaded network. Here, instead of connecting all the feature mapping nodes to the enhancement nodes, only the last group of feature mapping nodes is connected with the enhancement nodes.

Similarly, for the given input data X, the network with n groups of feature nodes and m groups of enhancement nodes is formulated as follows:

$$Y = [Z^n|H^m]W_n^m$$

where

$$\begin{aligned} Z_k &\triangleq \phi^k\left(X; \{W_{e_i}, \beta_{e_i}\}_{i=1}^k\right), \quad \text{for } k = 1, \ldots, n \\ H_j &\triangleq \xi(Z_nW_{h_j} + \beta_{h_j}), \quad \text{for } j = 1, \ldots, m \\ Z^n &\triangleq [Z_1, \ldots, Z_n] \\ H^m &\triangleq [H_1, \ldots, H_m] \end{aligned}$$

and $W_n^m = [Z^n|H^m]^+Y$. The matrix of the connecting weights W_n^m is calculated by the ridge regression directly, and the structure of the network is shown in Fig. 4(a).

Typically, the cascade of feature mapping [see (14)] is similar to the definition of recurrent system, which is very efficient in modeling sequential data. The recurrent structure is perfect for text document understanding and time series processing that deal with timing information in the input.

The recurrent information can be modeled in the feature nodes as the *recurrent feature nodes* in the following [the structure is illustrated in Fig. 4(b)] in order to learn sequential information

$$Z_k = \phi(Z_{k-1}W_{e_k} + XW_{z_k} + \beta_{e_k}), \quad p = 1, \ldots, n$$

where the matrices W_{z_k}, W_{e_k}, and β_{e_k} are randomly generated. Specifically, in the recurrent model, each Z_k is computed under the previous feature Z_{k-1} and the input X simultaneously. Based on this variant, a recurrent-BLS and long short-term memory-BLS can be constructed. The experiments for 12 real-world natural language processing classification data sets from CrowdFlower are reported in [19], and the proposed models achieve much better results than the benchmark methods in both accuracy and training time.

Remark: The structure of the proposed network in this part leads to new enhancement nodes if the feature nodes are incrementally added. Hence, only the increment of the additional enhancement nodes is available here, and the algorithm is similar to the corresponding section of the original BLS in [3]. Therefore, the details are ignored here.

C. Broad Learning Systems: Cascade of Enhancement Nodes (CEBLS) or Recurrent Enhancement Nodes

This proposed BLS model reconstructs the enhancement nodes by cascade of function composition. Again, for the input data X, the first n groups of feature nodes are generated by the following equations:

$$Z_i \triangleq \phi(XW_{e_i} + \beta_{e_i}), \quad i = 1, \ldots, n$$

and W_{e_i} and β_{e_i} are sampled from the given distribution. Project the feature nodes $Z^n \triangleq [Z_1, \ldots, Z_n]$ by function $\xi(\cdot)$, we have that the first group of enhancement nodes is

$$H_1 \triangleq \xi(Z^nW_{h_1} + \beta_{h_1}) \triangleq \xi(Z^n; \{W_{h_1}, \beta_{h_1}\})$$

where the associated weights are randomly sampled. The second group of enhancement nodes H_2 is compositely

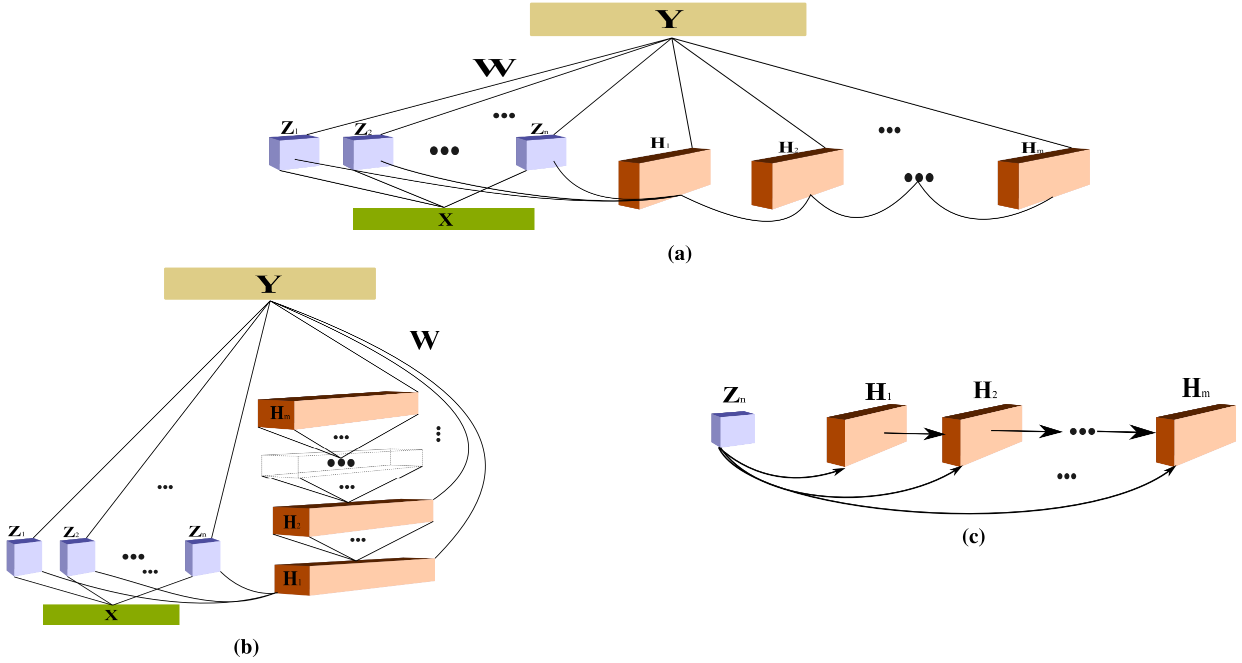


Fig. 5. CEBLS. Cascade of enhancement nodes. (a) Broad structure. (b) Structure redrawn, as right side showing the cascade architecture. (c) Alternative enhancement nodes connection (recurrent structure).

established as follows:

$$\begin{aligned} \boldsymbol{H}_2 &= \xi(\boldsymbol{H}_1 \boldsymbol{W}_{h_2} + \boldsymbol{\beta}_{h_2}) \\ &= \xi(\xi(\boldsymbol{Z}^n \boldsymbol{W}_{h_1} + \boldsymbol{\beta}_{h_1})\boldsymbol{W}_{h_2} + \boldsymbol{\beta}_{h_2}) \\ &\triangleq \xi^2(\boldsymbol{Z}^n; \{\boldsymbol{W}_{h_i}, \boldsymbol{\beta}_{h_i}\}_{i=1,2}). \end{aligned}$$

Furthermore, the first m groups of enhancement nodes are

$$\boldsymbol{H}_u \triangleq \xi^u(\boldsymbol{Z}^n; \{\boldsymbol{W}_{h_i}, \boldsymbol{\beta}_{h_i}\}_{i=1}^u), \quad \text{for } u = 1, \ldots, m \tag{18}$$

where $\boldsymbol{W}_{h_i}$ and $\boldsymbol{\beta}_{h_i}$ are randomly generated under the given distribution.

Consequently, the nodes $\boldsymbol{Z}^n$ and $\boldsymbol{H}^m \equiv [\boldsymbol{H}_1, \ldots, \boldsymbol{H}_m]$ are connected directly with the output, and the modified BLS is

$$\boldsymbol{Y} = [\boldsymbol{Z}^n | \boldsymbol{H}^m] \boldsymbol{W}_n^m$$

and $\boldsymbol{W}_n^m$ is calculated through the pseudoinverse of $[\boldsymbol{Z}^n | \boldsymbol{H}^m]$.

Next, the incremental learning algorithm for cascade of enhancement nodes is detailed in the following. Suppose that the $(n+1)$th set of feature nodes is incrementally added as $\boldsymbol{Z}_{n+1} \triangleq \phi(\boldsymbol{X}\boldsymbol{W}_{e_{n+1}} + \boldsymbol{\beta}_{e_{n+1}})$. The uth group of enhancement nodes should be supplemented by $\xi^u(\boldsymbol{Z}_{n+1}; \{\boldsymbol{W}_{\text{ex}_i}, \boldsymbol{\beta}_{\text{ex}_i}\}_{i=1}^u)$, $u = 1, \ldots, m$ and the corresponding matrix is denoted as

$$\boldsymbol{H}_{\text{ex}_m} \triangleq \Big[\xi(\boldsymbol{Z}_{n+1}; \{\boldsymbol{W}_{\text{ex}_1}, \boldsymbol{\beta}_{\text{ex}_1}\}), \\ \ldots, \xi^m\big(\boldsymbol{Z}_{n+1}; \{\boldsymbol{W}_{\text{ex}_i}, \boldsymbol{\beta}_{\text{ex}_i}\}_{i=1}^m\big)\Big]$$

where $\boldsymbol{W}_{\text{ex}_i}, \boldsymbol{\beta}_{\text{ex}_i}, i = 1, \ldots, m$ are randomly generated.

Next, the $(m+1)$th group of enhancement nodes is formulated as

$$\boldsymbol{H}_{m+1} \triangleq \xi^{m+1}\big(\boldsymbol{Z}^{n+1}; \{\boldsymbol{W}_{h_i}, \boldsymbol{\beta}_{h_i}\}_{i=1}^{m+1}\big)$$

where $\boldsymbol{Z}^{n+1} \triangleq [\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{n+1}]$, and $\boldsymbol{W}_{h_{m+1}}$ and $\boldsymbol{\beta}_{h_{m+1}}$ are randomly sampled. Therefore, the matrix $\boldsymbol{A}_n^m \triangleq [\boldsymbol{Z}^n | \boldsymbol{H}^m]$ is updated as $\boldsymbol{A}_{n+1}^{m+1} \triangleq [\boldsymbol{A}_n^m | \boldsymbol{Z}_{n+1} | \boldsymbol{H}_{\text{ex}_m} | \boldsymbol{H}_{m+1}]$. In fact, the output weights $\boldsymbol{W}_{n+1}^{m+1}$ could be dynamically updated under (15)–(17) since the notations of $\boldsymbol{A}_n^m$ and $\boldsymbol{A}_{n+1}^{m+1}$ are actually equivalent. The flatted network is illustrated in Fig. 5(a). Fig. 5(b) is the redrawn illustration of the flatted network, where the enhancement nodes in the right side are redrawn in a deep way.

Similar to the last section, the cascade enhancement nodes [see (18)] could be reconstructed in the form of recurrent. In order to capture the dynamic characteristics of the data, the enhancement nodes are recurrent connected and computed based on the previous enhancement nodes and feature nodes simultaneously. Therefore, for the given transition function ξ, the *recurrent enhancement nodes* [the structure is illustrated in Fig. 5(c)] are formulated as

$$\boldsymbol{H}_j = \xi(\boldsymbol{H}_{j-1}\boldsymbol{W}_{h_j} + \boldsymbol{Z}^n \boldsymbol{W}_{z_j} + \boldsymbol{\beta}_{h_j}), \quad j = 1, \ldots, m$$

where $\boldsymbol{W}_{z_j}$ is the added weights for the features $\boldsymbol{Z}^n$. To test the performance of the variants in time series, experiments on two typically chaotic systems are provided in [20]. The prediction accuracy for the given benchmark data sets is significantly improved and outperforms other models.

D. Broad Learning Systems: Cascade of Feature Mapping Nodes and Enhancement Nodes (CFEBLS)

This section takes a comprehensive cascade of both feature mapping nodes and enhancement nodes. Again for given input data set $\boldsymbol{X}$, and output data $\boldsymbol{Y}$, the composite feature nodes $\boldsymbol{Z}_k, k = 1, \ldots, n$, are generated by

$$\boldsymbol{Z}_k \triangleq \phi^k\big(\boldsymbol{X}; \{\boldsymbol{W}_{e_i}, \boldsymbol{\beta}_{e_i}\}_{i=1}^k\big)$$

where the weights are randomly sampled. Then, the m groups of enhancement nodes are generated as

$$\boldsymbol{H}_u \triangleq \xi^u\big(\boldsymbol{Z}^n; \{\boldsymbol{W}_{h_i}, \boldsymbol{\beta}_{e_i}\}_{i=1}^u\big), \quad \text{for } u = 1, \ldots, m$$

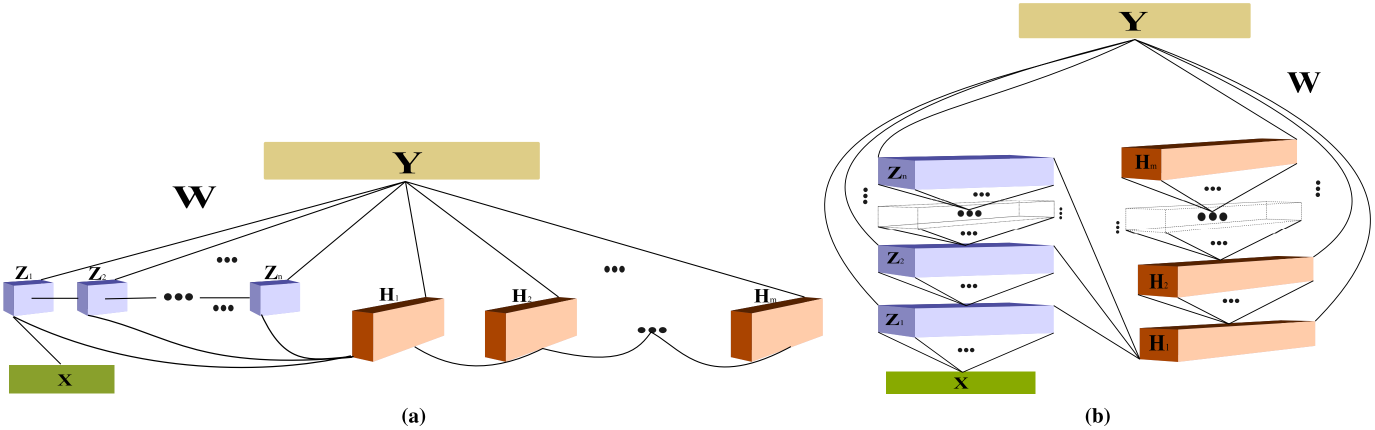


Fig. 6. CFEBLS. Comprehensive cascade composition model. (a) Broad structure. (b) Structure redrawn.

and all the associated weights are sampled under a specific distribution.

Consequently, the network could be formulated as

$$Y = [Z^n|H^m]W_n^m$$

where

$$Z^n \triangleq [Z_1, \ldots, Z_n]$$
$$H^m \triangleq [H_1, \ldots, H_m]$$

and $W_n^m = [Z^n|H^m]^+Y$.

Regarding the incremental learning algorithm for the increment of additional feature nodes and enhancement nodes, the matrix $A_n^m \triangleq [Z^n|H^m]$ is updated to $A_{n+1}^{m+1} \triangleq [A_n^m|Z_{n+1}|H_{\text{ex}_m}|H_{m+1}]$, where

$$\begin{aligned} Z_{n+1} &\triangleq \phi^{n+1}\big(X; \{W_{e_i}, \beta_{e_i}\}_{i=1}^{n+1}\big) \\ H_{\text{ex}_m} &\triangleq \big[\xi(Z_{n+1}; \{W_{\text{ex}_1}, \beta_{\text{ex}_1}\}), \\ &\quad \ldots, \xi^m\big(Z_{n+1}; \{W_{\text{ex}_i}, \beta_{\text{ex}_i}\}_{i=1}^m\big)\big] \\ H_{m+1} &\triangleq \xi^{m+1}\big(Z^{n+1}; \{W_{h_i}, \beta_{h_i}\}_{i=1}^{m+1}\big) \\ Z^{n+1} &\triangleq [Z_1, \ldots, Z_{n+1}] \\ H^{m+1} &\triangleq [H_1, \ldots, H_{m+1}] \end{aligned}$$

where the weights $W_{e_{n+1}}, \beta_{e_{n+1}}$ and $\{W_{h_i}, \beta_{h_i}\}_{i=1}^{m+1}$ are randomly sampled and fixed. Finally, the flatted network of this cascade structure is illustrated in Fig. 6(a) and an identical deep representation is redrawn in Fig. 6(b).

Remark: Alternative structure of BLS with all cascade groups of feature mapping nodes connected to all the cascade of enhancement nodes is also available. However, the equations of the network are essentially similar to the proposed ones, and details are omitted here.

E. Broad Learning Systems: Composite Model Versus Wide and Deep Learning

Recently, wide and deep learning has been discussed ([21], [22]), where the structure combines a single-layer linear systems and deep neural networks. This model is similar to our structure (see Fig. 5(a) and (b)) in Section IV-C.

Although the original BLS is designed in the form of flatted neural network, the cascade models proposed in this paper can be redrawn in deep structure, equivalently, in Figs. 3(b), 5(b), and 6(b), respectively. These redrawn models are not only broad but also "deep."

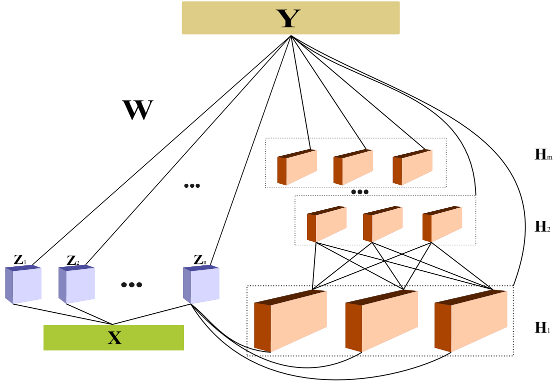


Fig. 7. CEBLS. Modified network for cascade of enhancement nodes.

Recall that an original BLS with n groups of feature maps and m groups of enhancement nodes is shown in Fig. 1. If the weights that connect the first $n-1$ groups of feature maps and the first group of enhancement nodes are enforced to be 0, the modified system is essentially the same with the model in [21] except for the full-link connection between the $m+n$ groups of nodes in BLS and the output layer. This modified network is redrawn in Fig. 7.

F. Broad Learning Systems: Cascade of Convolution Feature Mapping Nodes (CCFBLS)

CNN has been a capable tool for pattern recognition if the weights between the layers are chosen appropriately. Fig. 8 can be considered a BLS model with cascade of convolution functions, where feature mapping nodes are the nodes going through the mapping by convolution and pooling operators. In another words, this model is a specific case of CFBLS (see Section IV-A), which is a cascade of convolution feature mapping nodes (CCFBLS).

The network based on convolutional functions is constructed under the cascade of convolution and pooling operations in the feature mapping nodes. First, the feature mapping nodes $\phi(\cdot)$ are defined as follows:

$$\begin{aligned} Z_k &= \phi(Z_{k-1}; \{W_{e_k}, \beta_{e_k}\}) \\ &\triangleq \theta(P(Z_{k-1} \otimes W_{e_k} + \beta_{e_k})), \quad \text{for } k = 1, \ldots, n \end{aligned}$$

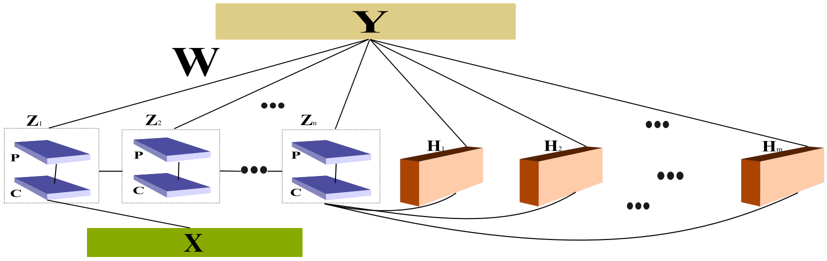


Fig. 8. CCFBLS. Cascade of Convolution feature mapping nodes.

where the operator $\otimes$ is the convolutional function for the given matrices, the function $P(\cdot)$ is the pooling operation, and $\theta(\cdot)$ is the selected activation function. Furthermore, the weights of convolutional filters are randomly sampled under a given distribution. Second, the expected network is enhanced by function $\boldsymbol{H}_j \triangleq \xi(\boldsymbol{Z}^n \boldsymbol{W}_{h_j} + \boldsymbol{\beta}_{h_j})$, for $j = 1, \ldots, m$, where $\boldsymbol{Z}^n \triangleq [\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n]$. Finally, to ensure as much as information passed into the output layer, $\boldsymbol{Z}^n$ and $\boldsymbol{H}^m$ are connected directly with the desired $\boldsymbol{Y}$. The whole network is illustrated in Fig. 8. This architecture can be considered as a variant of a 3-D CNN, where the connections are established from every layer to the output layer. This composite model has been tested on CIFAR-10 and CIFAR-100 data sets. The results are very promising and outperform existing models in accuracy and time [23].

G. Fuzzy Model in the Feature Nodes: Fuzzy Broad Learning System

The Takagi–Sugeno (TS) fuzzy system can be merged into BLS to form the fuzzy BLSs. The fuzzy BLS replaces the feature nodes of BLS with a group of TS fuzzy subsystems. The outputs of fuzzy rules produced by every fuzzy subsystem in the feature nodes are sent to the enhancement layer for further nonlinear transformation to preserve the characteristic of inputs. Details of fuzzy BLS can be found in [24].

V. Experiments

We discussed and formulated different variants of BLS in Section IV with cascade of feature mapping nodes and/or enhancement nodes. These frameworks have been tested on different data sets, and additional applications are developed and submitted in [19], [20], and [24]. In this section, comparisons of different data sets between the default BLS and its variants are given.

In the following experiments, only the original BLS is used to compare with typical models, such as SVM, LSSVM, and ELM, on some representative benchmarks for function approximation, time series prediction, and face recognition.

A. Function Approximation

1) UCI Data Sets for Regression: We select 10 regression data sets from the University of California, Irvine (UCI) database [25], which fall into three categories: small size and low dimensions, medium size and dimensions, and large size and low dimensions. The details of the data sets are listed in Table I.

The cost parameter C and kernel parameter γ of SVM, LSSVM [26], and ELM [27] play an important role in learning a good regression model, hence they have to be chosen appropriately for a fair comparison. In this paper, we carry out a grid search for the parameters (C, γ) from $\{2^{-24}, 2^{-23}, \ldots, 2^{24}, 2^{25}\}$ to determine the optimal settings for SVM (using *libsvm* [28]) and ELM, whereas the optimal values of (C, γ) for LSSVM are decided by itself using *LS-SVMlab* Toolbox. We also perform a grid search for the parameters of BLS, including the numbers of feature nodes N_f, mapping groups N_m, and enhancement nodes N_e from $[1, 10] \times [1, 30] \times [1, 200]$, and the searching step is 1. The parameter settings of the above-mentioned models are shown in Table II.

TABLE I

Details of Data Sets for Regression

Data set	No. of samples		Input variables
	Training	Testing	
Abalone	2784	1393	8
Basketball	64	32	4
Bodyfat	168	84	14
Cleveland	202	101	13
Housing	337	169	13
Mortgage	699	350	15
Pyrim	49	25	27
Quake	1452	726	3
Strike	416	209	6
Weather Izmir	974	487	9

We choose the best results from 10 trials for each data set, and the root-mean-squared errors (RMSE) of SVM, LSSVM, ELM, and BLS are given in Table III.

It can be concluded that the BLS outperforms the SVM, LSSVM, and ELM in testing accuracy on the 10 function approximation data sets.

B. Time Series Prediction

We use a wind speed data set [29] to compare the performance of BLS with autoregression (AR), adaptive network-based fuzzy inference system (ANFIS), SVM, and predictive deep Boltzmann machines (PDBMs) [29] on predicting the short-term wind speed. There are 50 000 wind speed data recorded in every 10 min for training and 2500 for testing. In the experiment, the look-back interval is 100 min, i.e., we use the last 10 wind speeds to predict the next one. The models are first trained by all the 50 000 samples, and they are then employed to forecast the wind speed in 10 min–2 h ahead based on the testing data. The mean absolute percentage error is adopted to evaluate the models involved. The parameters for BLS are $N_f = 10$, $N_m = 14$, and $N_e = 440$.

TABLE II

PARAMETER SETTINGS OF SVM, LSSVM, ELM, AND BLS FOR UCI DATA SETS

Data set	SVM		LSSVM		ELM		BLS		
	C	γ	C	γ	C	γ	N_f	N_m	N_e
Abalone	2^2	2^{-1}	2.8932	3.0774	2^0	2^0	5	6	41
Basketball	2^0	2^0	6.0001	27.3089	2^{25}	2^{11}	6	7	4
Bodyfat	2^2	2^{-2}	6505.7167	233.8448	2^{14}	2^5	6	5	13
Cleveland	2^2	2^2	0.7527	45.2507	2^{13}	2^{15}	1	10	9
Housing	2^2	2^1	61.9215	6.4770	2^6	2^0	5	29	50
Mortgage	2^{10}	2^{-1}	803.9607	5.4985	2^{13}	2^3	9	4	135
Pyrim	2^{10}	2^8	52.5877	3.2463	2^2	2^6	3	7	2
Quake	2^5	2^5	0.1115	0.0079	2^5	2^{14}	10	2	6
Strike	2^0	2^{-4}	0.3167	0.7383	2^{-1}	2^5	9	11	30
Weather Izmir	2^4	2^{-2}	1433.0467	44.2146	2^{12}	2^2	4	3	87

TABLE III

RMSE COMPARISON OF SVM, LSSVM, ELM, AND BLS ON DATA SETS FOR FUNCTION APPROXIMATION

Data set	SVM		LSSVM		ELM		BLS	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Abalone	0.0748	0.0773	0.0717	0.0756	0.0756	0.0777	0.0737	**0.0754**
Basketball	0.0804	0.0767	0.0826	0.0744	0.0801	0.0719	0.0834	**0.0659**
Bodyfat	0.0038	0.0049	0.0027	0.0038	0.0042	0.0033	0.0060	**0.0030**
Cleveland	0.1032	0.1514	0.1039	0.1256	0.1038	0.1281	0.1040	**0.1199**
Housing	0.0286	0.0792	0.0282	0.0780	0.0371	0.0762	0.0571	**0.0751**
Mortgage	0.0019	0.0046	0.0026	0.0053	0.0042	0.0057	0.0031	**0.0043**
Pyrim	0.0130	0.1549	0.0225	0.1133	0.0767	0.1376	0.0420	**0.0578**
Quake	0.1603	0.2029	0.1576	0.1733	0.1716	0.1729	0.1703	**0.1718**
Strike	0.0584	0.1053	0.0463	0.1007	0.0592	0.1043	0.0503	**0.1001**
Weather Izmir	0.0166	0.0190	0.0161	0.0191	0.0158	0.0191	0.0165	**0.0188**

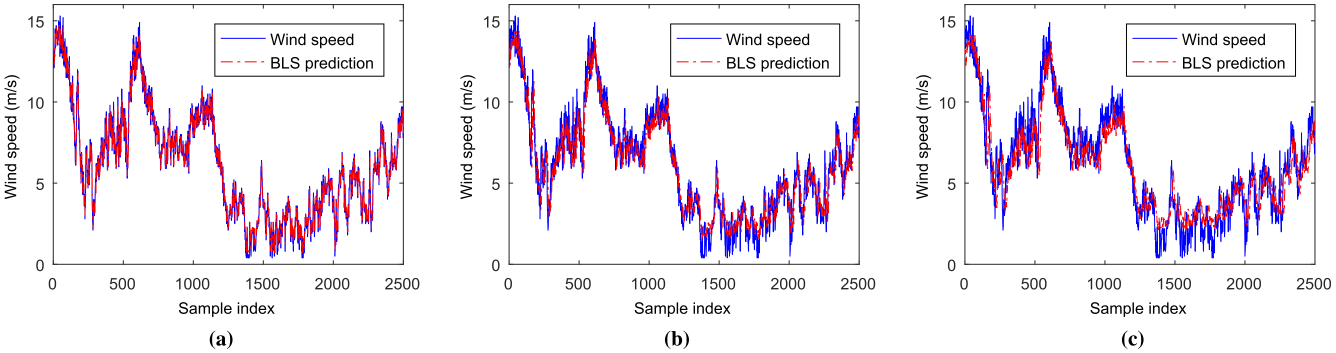


Fig. 9. Wind speed prediction of BLS for (a) 10 min ahead, (b) 60 min ahead, and (c) 120 min ahead.

The prediction results for 10, 60, and 120 min ahead of BLS are shown in Fig. 9. We can see that the BLS has the relatively small AE, and the AEs of other models behave some high oscillations.

The performance of the models is compared in Table IV, and it is obvious that the BLS achieves the best prediction accuracy on short-term wind speed forecasting.

C. Face Recognition

In this section, we select three popular face data sets, including the *Extend YaleB* [30], *The ORL Database of Faces (ORL)* [31], and *The University of Manchester Institute of Science and Technology (UMIST)* [32], to compare the classification ability of BLS with SVM, LSSVM, and ELM. The details of the above-mentioned face data sets are as follows.

1) Extended YaleB: The extended YaleB face database comprises 2414 cropped images of 38 subjects with the size of 32 × 32 pixels. The images have large variations in terms of illumination conditions and expressions for each subject. There are 30 images of each person for training and the remain 1274 images for testing.

2) ORL: The ORL data set consists of 400 gray-scaled face images of 40 different persons with the size of 32 × 32 taken between April 1992 and April 1994 at AT&T Laboratories Cambridge, and there are 10 different images of each person. We randomly choose 6 pictures of each person for training set and the rest 160 pictures for testing.

3) UMIST: The UMIST face database is composed of 575 images of 20 distinct subjects with resolution 112 × 92. This data set is more challenging because of the larger variations between the images of the same face in viewing direction than regular image variations in face identity. We resize them to 56 × 46 and randomly select 15 images per subject for training and the rest for testing.

TABLE IV
COMPARISON OF PERFORMANCE ON SHORT-TERM WIND SPEED PREDICTION

Min. ahead	MAPE$\times 10^{-2}$				
	AR	ANFIS	SVM	PDBM	BLS
10	7.57	10.84	14.55	7.05	**6.87**
20	9.68	14.96	17.01	8.80	**8.65**
30	11.19	18.01	19.04	10.12	**9.94**
40	12.55	20.49	21.03	11.40	**11.07**
50	13.71	22.20	22.19	12.25	**11.87**
60	14.78	23.60	23.35	12.98	**12.61**
70	15.74	25.12	24.96	13.69	**13.31**
80	16.58	26.38	25.93	14.16	**13.84**
90	17.41	27.61	27.05	14.81	**14.29**
100	18.15	28.64	27,29	15.48	**15.08**
110	18.95	29.69	27.32	16.01	**15.63**
120	19.73	30.48	28.64	16.63	**16.12**

Grid search method is also applied to find the optimal settings of parameters for these models, and the searching intervals for SVM and ELM are set to be the same as earlier. The searching range for BLS is expended to $[1, 60] \times [1, 50] \times [1, 6000]$. The parameter settings are shown in Table V, and the classification results are listed in Table VI.

We can see that the BLS always has the highest classification accuracies in recognizing faces from the three data sets, which confirms that it outperforms the popular discriminative models.

D. Classification for BLS's Variants

In this section, the MNIST data set [33] and NORB data set [4] are chosen to compare the classification abilities of BLS and its variants.

1) MNIST: The MNIST data set consists of 60 000 training images and 10 000 testing images of 10 classes with the size of 28×28. Meanwhile, considering the length limitation of this paper, the incremental algorithms of the variants are not shown in this section, i.e., only the one-shot versions are considered and compared.

2) NORB: NORB data set consists of 48 600 images with $2 \times 32 \times 32$ pixels each. The objects in the data set belong to five distinct classes that are: 1) animals; 2) humans; 3) airplanes; 4) trucks; and 5) cars. Among them, 24 300 images are selected for training and the other 24 300 images are for testing.

Since the classification of MNIST data and NORB data is not challenging, the cascade of the feature maps and the enhancement nodes is set as 2 consistently. We perform a grid search for the associated parameters, including the following sets: 1) the numbers of two-layer-cascaded feature nodes N_{fc}, the mapping groups N_{mc}, and the enhancement nodes N_e for the cascade of the feature mapping nodes (CFBLS); 2) the numbers of two-layer-cascaded feature nodes N_{flc}, the mapping groups N_{mlc}, and the enhancement nodes N_e for the limited connection between the groups of cascaded feature maps and the enhancement nodes (LCFBLS); 3) the numbers of feature nodes N_f, the mapping groups N_m, and the two-layer-cascaded enhancement nodes N_{ec}, the cascade of the enhancement nodes (CEBLS); and 4) the numbers of two-layer-cascaded feature nodes N_{flc}, the mapping groups N_{mlc}, and the two-layer-cascaded enhancement nodes N_{elc} for Cascade of feature mapping nodes and enhancement nodes (CFEBLS). To simplify the denotations of this section, the number of feature nodes, the mapping groups, and the enhancement nodes are set uniformly as N_f, N_m, and N_e, respectively.

The classification results of MNIST and NORB are listed in Tables VII and VIII, respectively. We can observe that the accuracy results of the MNIST and NORB are actually in the same level. This may be caused by its favorable properties. However, the structure of the networks differs from each other, which implies that the cascade variants outperforms the default BLS in structure optimization and most of case, less number of parameters. Again, other applications are developed and submitted in [19], [20], and [24].

E. Face Recognition Comparison in CCFBLS and Resnet-34

The BLS composite model using convolution feature nodes, CCFBLS, shown in Section IV-F and Fig. 8, is used to test on one of the most challenging large-scale face recognition databases, MS-Celeb-1M [34]. This data set is designed as a benchmark task of face recognition to: 1) recognize one million celebrities from the face images and link them to the corresponding entity keys in a knowledge base [34] and 2) investigate low-shot face recognition with the goal to build a large-scale face recognizer capable of recognizing a substantial number of individuals with high precision and high recall [35]. Some images from the MS-Celeb-1M data sets are depicted in Fig. 10.

The details of the experiment are designed as follows. The original data set consists of 21 000 persons each with 50–100 images. This data set is divided into two sets, 20 000 persons in the base set and the rest 1000 persons in novel set. In the base set, tens of images for each celebrity are given to train the face representation model, while in the novel set, only one image is provided to train the model. In our experiment, the images for the first 2000 persons in the base set are selected to test the proposed CCFBLS.

The selected subset of MS-Celeb-1M contains 119 134 color images associated with 2000 persons for training and 10 000 color images for testing. The average size of images is $250 \times 300 \times 3$ pixels. This is considered as a very large size of image compared with Extended YaleB, ORL, and UMIST databases that only 32×32 pixels are used, and this surely increases the complexity and difficulty of the learning.

To demonstrate the effectiveness and efficient of the proposed CCFBLS, the result is compared with the residual network, which has been proven to be a popular and powerful tool in image processing and recognition. The standard residual network with 34 layers (Resnet-34) [36] without special feature tricks is constructed. The CCFBLS is constructed with only 18 convolution functions, and four of the convolution outputs are connected to the output nodes of the CCFBLS. Both simulations are performed on a machine using an Intel Corei7-7800X with a NVIDA GeForce GTX1080TICUDA.

TABLE V
PARAMETER SETTINGS OF SVM, LSSVM, ELM, AND BLS FOR FACE DATA SETS

Data set	SVM		LSSVM		ELM		BLS		
	C	γ	C	γ	C	γ	N_f	N_m	N_e
Extended YaleB	2^{13}	2^{-14}	6.4732	1628.5719	2^{13}	2^{11}	60	30	6000
ORL	2^{4}	2^{-23}	6.4993	13491.7042	2^{6}	2^{21}	26	10	460
UMIST	2^{6}	2^{-11}	6.5013	425.3081	2^{5}	2^{8}	10	9	575

TABLE VI
CLASSIFICATION ACCURACIES OF SVM, LSSVM, ELM, AND BLS ON FACE DATA SETS

Data set	SVM		LSSVM		ELM		BLS	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Extended YaleB	99.30%	90.89%	98.95%	88.15%	99.56%	96.94%	100%	**97.65%**
ORL	100%	94.38%	100%	82.50%	100%	96.25%	100%	**97.50%**
UMIST	100%	96.36%	100%	92.00%	100%	96.73%	100%	**98.18%**

TABLE VII
COMPARISON OF PERFORMANCE BETWEEN BLS'S VARIATIONS ON MNIST DATA SET

Algorithms	No. of Nodes or Groups			No. of Para.	Accuracy
	N_f	N_m	N_e		
BLS	10	10	11000	111k	98.72%
CFBLS	10	4	7800	78.8k	98.76%
LCFBLS	6	10	8000	80.6k	98.74%
CEBLS	10	9	4300	86.9k	98.79%
CFEBLS	8	5	3600	72.8k	**98.83%**

TABLE VIII
COMPARISON OF PERFORMANCE BETWEEN BLS'S VARIATIONS ON NORB DATA SET

Algorithms	No. of Nodes or Groups			No. of Para.	Accuracy
	N_f	N_m	N_e		
BLS	10	100	9000	50k	89.06%
CFBLS	10	86	6300	40.1k	89.43%
LCFBLS	10	50	7300	39k	89.54%
CEBLS	7	90	3600	39.15k	89.88%
CFEBLS	8	50	3800	42k	**90.02%**

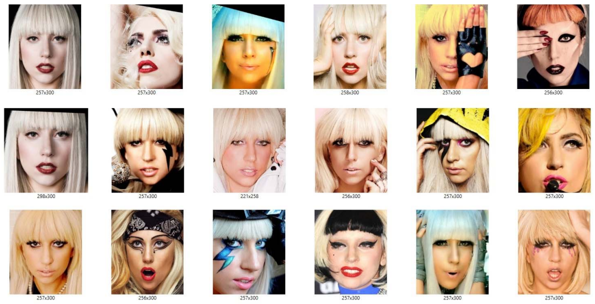


Fig. 10. Some face images from the MS-Celeb-1M, and the example training samples are selected from base set [35].

TABLE IX
PERFORMANCE COMPARISON BETWEEN CCFBLS AND RESNET-34 ON MS-CELEB-1M FACE DATABASE

Algorithms	No. of neurons.	No. of para.	Training time	Accuracy
Resnet-34	188880	23.794Mil	216min	90.72%
CCFBLS	**108196**	**12.290Mil**	**123min**	**91.96%**

The environment of the experiments was Linux Ubuntu, with Tensorflow 1.7.0 installed.

Detailed results are shown in Table IX. In CCFBLS, the total of 108 196 neurons are needed, whereas in Resnet-34, the total number of neurons are 188 880. Total of 12.290 million parameters are used in CCFBLS, whereas 23.794 million parameters are used in Resnet-34. It is shown that almost half of the neurons, parameters, and half of the training time are needed for the proposed CCFBLS to reach a better testing accuracy.

VI. CONCLUSION

The frameworks of different variations of BLS structure are proposed. This kind of establishment is to offer alternatives of constructing flatted networks for future research. The incremental learning algorithms of original BLS can still apply to these variants, where weight connections are established within the feature mapping nodes, within enhancement nodes, or between feature nodes and enhancement nodes. Mathematical modeling of these variants is also given. Several deep and wide neural networks [10], [12], [13] can be considered as a special arrangement of the proposed BLS variants.

Adapting proofs from Hornik, we also prove that BLS is a universal function approximator, which states that any measurable function on $\mathbb{R}^d$ can be approximated arbitrarily well by a BLS with nonconstant bounded feature mapping and activation function in μ measure.

To test the approximation capability, the regression performance of BLS is compared with SVM, LSSVM, and ELM on UCI database and face recognition data sets, including Extend YaleB, ORL, and UMIST. Similarly, AR, ANFIS, SVM, and PDBM approaches are compared with BLS on time series prediction. In order to have a fair comparison, parameters that can achieve the best testing accuracy are generated through a grid search for all the approaches. After that the classification ability of variants of BLS is tested in NORB and MNIST. It is shown that the proposed Cascade Convolution Feature mapping nodes BLS (CCFBLS) variant can achieve a better testing recognition accuracy with only almost half of the neurons, the parameters, and the training time in compared with Resnet-34 structure in MS-Celeb-1M large-scale image data set. It is shown that for given the benchmark data, the BLS and its variants outperform the above-mentioned algorithms in testing accuracy.

References

[1] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[2] Y. LeCun *et al.*, “Handwritten digit recognition with a back-propagation network,” in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.

[3] C. L. P. Chen and Z. L. Liu, “Broad learning system: An effective and efficient incremental learning system without the need for deep architecture,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018, doi: 10.1109/TNNLS.2017.2716952.

[4] Y. LeCun, F. J. Huang, and L. Bottou, “Learning methods for generic object recognition with invariance to pose and lighting,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Jun. 2004, pp. 97–104.

[5] M. Klassen, Y. Pao, and V. Chen, “Characteristics of the functional link net: A higher order delta rule net,” in *Proc. IEEE Int. Conf. Neural Netw.*, Jul. 1988, pp. 507–513.

[6] Y.-H. Pao, S. M. Phillips, and D. J. Sobajic, “Neural-net computing and the intelligent control of systems,” *Int. J. Control*, vol. 56, no. 2, pp. 263–289, 1992.

[7] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[8] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.

[9] B. Igelnik and Y.-H. Pao, “Stochastic choice of basis functions in adaptive function approximation and the functional-link net,” *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.

[10] S. Dehuri and S.-B. Cho, “A comprehensive survey on functional link neural networks and an adaptive PSO–BP learning for CFLNN,” *Neural Comput. Appl.*, vol. 19, no. 2, pp. 187–205, 2010.

[11] C. L. P. Chen, “A rapid supervised learning neural network for function interpolation and approximation,” *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1220–1230, Sep. 1996.

[12] C. L. P. Chen, S. R. LeClair, and Y.-H. Pao, “An incremental adaptive implementation of functional-link processing for function approximation, time-series prediction, and system identification,” *Neurocomputing*, vol. 18, nos. 1–3, pp. 11–31, 1998.

[13] C. L. P. Chen and J. Z. Wan, “A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 1, pp. 62–72, Feb. 1999.

[14] W. Rudin, *Real and Complex Analysis* (Higher Mathematics Series). New York, NY, USA: McGraw-Hill, 1987.

[15] P. Guo, M. R. Lyu, and C. L. P. Chen, “Regularization parameter estimation for feedforward neural networks,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 1, pp. 35–44, Feb. 2003.

[16] Z. Xu, X. Chang, F. Xu, and H. Zhang, “$l_1/2$regularization: A thresholding representation theory and a fast solver,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1013–1027, Jul. 2012.

[17] J. W. Jin, Z. L. Liu, and C. L. P. Chen, “Discriminative graph regularized broad learning system for image recognition,” *Sci. China Inf. Sci.*, to be published, doi: 10.1007/s11432-017-9421-3.

[18] J. W. Jin, C. L. P. Chen, and Y. T. Li, “Regularized robust broad learning system for uncertain data modeling,” *Neurocomputing*, to be published.

[19] C. M. Vong, J. Du, and C. L. P. Chen, “Accurate and efficent text classification by simultaneous learning of multiple information using recurrent and gated broad learning system,” *IEEE Trans. Cybernatics*, to be published.

[20] M. L. Xu, M. Han, C. L. P. Chen, and T. Qiu, “Recurrent broad learning systems for time series prediction,” *IEEE Trans. Cybern.*, to be published.

[21] H.-T. Cheng *et al.*, “Wide & deep learning for recommender systems,” in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[22] G. Pandey and A. Dukkipati, “To go deep or wide in learning?” in *Proc. 17th Int. Conf. Articial Intell. Statist. (AISTATS)*, Reykjavik, Iceland, 2014, pp. 724–732.

[23] Q. Y. Feng, Z. L. Liu, and C. L. P. Chen, “Composite convolution kernels in broad learning systems for image recognition,” *IEEE Trans. Cybernatics*, to be published.

[24] S. Feng and C. L. P. Chen, “Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification,” *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2018.2857815.

[25] C. L. Blake and C. J. Merz, “UCI repository of machine learning databases,” Dept. Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, 1998, vol. 55. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html

[26] B. De Moor and J. Vandewalle, *Least Squares Support Vector Machines*. Singapore: World Scientific, 2002.

[27] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[28] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

[29] C.-Y. Zhang, C. L. P. Chen, M. Gan, and L. Chen, “Predictive deep Boltzmann machine for multiperiod wind speed forecasting,” *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1416–1425, Oct. 2015.

[30] K.-C. Lee, J. Ho, and D. Kriegman, “Acquiring linear subspaces for face recognition under variable lighting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.

[31] F. S. Samaria and A. C. Harter, “Parameterisation of a stochastic model for human face identification,” in *Proc. 2nd IEEE Workshop Appl. Comput. Vis.*, Dec. 1994, pp. 138–142.

[32] D. B. Graham and N. M. Allinson, “Characterising virtual eigensignatures for general purpose face recognition,” in *Face Recognition*. Berlin, Germany: Springer, 1998, pp. 446–456.

[33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[34] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “MS-Celeb-1M: A dataset and benchmark for large-scale face recognition,” in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016.

[35] Y. Guo and L. Zhang. (2017). “One-shot face recognition by promoting underrepresented classes.” [Online]. Available: https://arxiv.org/abs/1707.05574

[36] K. He, X. Zhang, S. Ren, and J. Sun. (2015). “Deep residual learning for image recognition.” [Online]. Available: https://arxiv.org/abs/1512.03385

C. L. Philip Chen (S’88–M’88–SM’94–F’07) received the Ph.D. degree from the University of Michigan at Ann Arbor, Ann Arbor, MI, USA.

He is currently a Chair Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China. Being a Program Evaluator of the Accreditation Board of Engineering and Technology Education (ABET), Baltimore, MD, USA, for computer engineering, electrical engineering, and software engineering programs, he successfully architects the University of Macau’s Engineering and Computer Science programs receiving accreditations from Washington/Seoul Accord through the Hong Kong Institution of Engineers (HKIE), Hong Kong, which is considered as his utmost contribution in engineering/computer science education for Macau as the former Dean of the Faculty. His current research interests include systems, cybernetics, and computational intelligence.

Dr. Chen is a fellow of The American Association for the Advancement of Science, The International Association for Pattern Recognition, the Chinese Association of Automation (CAA), and HKIE. He received the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater, Purdue University. He was the Chair of the TC 9.1 Economic and Business Systems of the International Federation of Automatic Control from 2015 to 2017. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS and an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and the IEEE TRANSACTIONS ON CYBERNETICS. From 2012 to 2013, he was the IEEE Systems, Man, and Cybernetics Society President. He is currently the Vice President of CAA.

Zhulin Liu received the bachelor's degree in mathematics from Shandong University, Jinan, Shandong, China, in 2009, and the M.S. degree in mathematics from the University of Macau, Macau, China, in 2011, where she is currently pursuing the Ph.D. degree with the Faculty of Science and Technology.

Her current research interests include computational intelligence, matching learning, and function approximation.

Shuang Feng received the B.S. degree in mathematics and the M.S. degree in applied mathematics from Beijing Normal University, Beijing, China, in 2005 and 2008, respectively. He is currently pursuing the Ph.D. degree in computer science with the Faculty of Science and Technology, University of Macau, Macau, China.

He is currently an Associate Professor with the School of Applied Mathematics, Beijing Normal University, Zhuhai, China. His current research interests include fuzzy systems, fuzzy neural networks, and their applications in computational intelligence.