International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

Analyzing the Universal Approximation Capabilities of Broad Learning Systems Across Structural Designs

Mr.Rachamalla Shyambabu
Assistant Professor, Department of CSE,
Malla Reddy College of Engineering for Women.,
Maisammaguda., Medchal., TS, India

Abstract

Here, a mathematical framework is presented that builds on the work done before to create a quick and effective discriminative broad learning system (BLS) by using flattened structure and incremental learning.

We give evidence that BLS has the universal approximation feature. In addition, a mathematical model is provided for the framework of numerous BLS versions. Different variants include cascade, recurrent, and broad-deep combination architectures.

On regression tasks involving function approximation, time series prediction, and face recognition, the BLS and its variants have been shown to beat a number of existing learning algorithms. We also provide trials on the MS-Celeb-1M data set, which is among the most difficult in the field. Comparisons with other convolutional networks show that BLS variations are superior in terms of both efficacy and efficiency.

Keywords: universal approximation, time-variant big data modeling, deep learning, face recognition, functional link neural networks (FLNNs), and broad learning system (BLS).

I. BACKGROUND

Recently, numerous freshly released machine learning techniques have helped to boost the resurgence of AI research.

The deep learning algorithm—which encompasses both generative and discriminative learning—is a crucial factor. The deep model of constrained Boltzmann machines [1] is an example of deep generative learning.

The convolutional neural network (CNN) [2] is another example of a discriminative learning system. Researchers in the field of machine learning may now follow the path blazed by these deep learning algorithms and models and their modifications. The learning algorithms have also been put to use in pattern recognition, picture recognition, voice recognition, and video processing, all of which have shown promising results.

The manuscript was received on October 25, 2017; it was edited on May 14, 2018 and again on July 13, 2018; it was approved on August 15, 2018. Some of the funding for this study came from the Macau Science and Technology Development Fund (Grants 019/2015/A1, 079/2017/A2, and 024/2015/AMJ), the Macau Natural Science Foundation (Grants 61751202, 61751205, and 61572540), and the University of Macau (Multiyear Research Grants). For further correspondence, please contact Zhulin Liu.

C. L. P. Chen is affiliated with the College of Navigation at Dalian Maritime University in Dalian, China (postal code 116026), the State Key Laboratory of Management and Control for Complex Systems at the Institute of Automation at the Chinese Academy of Sciences in Beijing, China (postal code 100080), and the IEEE (e-mail address: philip.chen@ieee.org).

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

Professor Z. Liu may be reached at zhulinlau@gmail.com or at the Faculty of Science and Technology, University of Macau, Macau 99999, China.

School of Applied Mathematics, Beijing Normal University, Zhuhai 519087, China; Faculty of Science and Technology, University of Macau, Macau 99999, China (e-mail: fengshuang@bnuz.edu.cn).

Some of the figures in this work have color counterparts that may be seen at http://ieeexplore.ieee.org.

Specific DOI: 10.1109/TNNLS.2018.2866622 .The manuscript was received on October 25, 2017; it was edited on May 14, 2018 and again on July 13, 2018; it was approved on August 15, 2018. Part of the funding for this study came from the National Natural Science Foundation of China (Grant No. 61751202).

Partially supported by the Macau Science and Technology Development Fund (Grant 019/2015/A1, Grant 079/2017/A2, and Grant 024/2015/AMJ), and the University of Macau (Multiyear Research Grants), Grant 61751205 and Grant 61572540 were awarded. For further correspondence, please contact Zhulin Liu.

C. L. P. Chen is affiliated with the College of Navigation at Dalian Maritime University in Dalian, China (postal code 116026), the State Key Laboratory of Management and Control for Complex Systems at the Institute of Automation at the Chinese Academy of Sciences in Beijing, China (postal code 100080), and the IEEE (e-mail address: philip.chen@ieee.org).

Professor Z. Liu may be reached at zhulinlau@gmail.com or at the Faculty of Science and Technology, University of Macau, Macau 99999, China.

School of Applied Mathematics, Beijing Normal University, Zhuhai 519087, China; Faculty of Science and Technology, University of Macau, Macau 99999, China (e-mail: fengshuang@bnuz.edu.cn).

Some of the figures in this work have color counterparts that may be seen at http://ieeexplore.ieee.org. Convolutional neural networks (CNNs) are a kind of multilayer neural networks that map features using convolution and pooling processes.

propriety with appropriately calibrated scales. Another way of looking at the feature mapping part is as a set of kernel mappings, whereby various kernels are substituted for the convolution and pooling operators. Numerous recognition contests using the ImageNet data set have shown that CNN and its derivatives are capable of a high recognition rate.

The deep structure has proved very effective, although the training procedure for most networks is lengthy due to the complexity of the underlying structures. High-performance computers and robust infrastructure are necessary for many of the research.

This additional complexity makes theoretical analysis of the deep structure very challenging, and as a result, most efforts are focused on tweaking the settings or adding layers to improve accuracy.

As of late, Chen and Liu [3] have created a rapid and effective discriminative learning system called broad learning system (BLS). When more nodes are required and input data is continually entering the neural networks, the designed neural networks extend the neural nodes extensively and update the weights of the neural networks gradually; this is done without stacking the layer-structure. Because of this, the BLS structure is ideal for modeling and learning in a large data setting where the variables change over time. Furthermore, results from the Modified National Institute of Standards and Technology (MNIST) and the handwriting recognition and New York University object recognition benchmark (NORB) databases [4] show that BLS significantly outperforms existing deep structures in learning accuracy and generalization ability.

Here, we present a mathematical evidence of BLS's universal approximation characteristic, in addition

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

to its excellent discriminative capabilities in classification and recognition.

In accordance with the theorem, BLS is defined as an approximator of nonlinear functions. This includes a comparison of BLS's regression performance to that of the support vector machine (SVM), the least squared SVM (LSSVM), and the extreme learning machine (ELM) on a number of benchmarked data sets for function approximation, time series prediction, and facial recognition.

Several BLS versions are also discussed in this study, each of which uses a unique set of weight connections between nodes in the feature mapping stage, between nodes in the enhancement stage, and between nodes in both stages. These versions are also modeled mathematically for your perusal. Future studies may benefit from these potentially applicable alternative structures.

After introducing the background material for this study, we prove the universal approximation feature of the BLS and then discuss several alternative

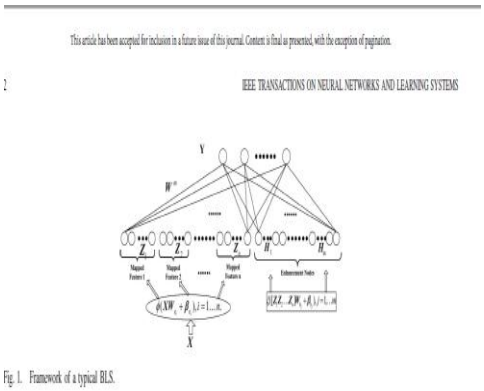


Fig. 1. Framework of a typical BLS.

Function approximation, time series prediction, and facial recognition are all areas where BLS structures and experiments have been used as yardsticks.

Experiments are performed to show the benefits of BLS variations.

the norm for large-scale data sets. Most difficult is the MS-Celeb-1M data set.

The findings demonstrate that BLS, a convolutional network with cascaded feature mapping nodes, performs better than competing networks.

B. INTRODUCTIONS

Synaptic Functional Link Neural Networks

It has been evolved further to the random vector functional link network [6] that the functional link neural network (FLNN) introduced by Klassen et al. [5] is a variation of the higher order neural network without hidden units. Due to its general approximation features, FLNN has spawned a wide variety of refinements, models, and successful applications (see [7]–[9]). For a thorough analysis of FLNN, see [10].

With the help of a supervised learning method called rank-expansion with immediate learning, Chen [11] and Chen et al. [12] introduced an adaptable implementation of the FLNN architecture. This fast approach is advantageous due to its ability to learn the weights in a single training phase without the need for repetitive parameter updates. Furthermore, in [13] a fast learning algorithm is proposed to find optimal weights of the flat neural networks (especially, the functional link network), which utilizes the linear least-square method, and this algorithm makes it simpler to update the weights instantly for incremental input patterns and enhancement nodes.

B. Generalized Studying Machines

Learning deep structures may be tedious due to the need to train a large number of parameters in the filters and layers, but the BLS [3] offers an alternate approach. In case the network is determined to have grown, it may be trained using incremental learning methods for rapid reshaping in widespread growth without requiring a retraining phase.

Using a flat network architecture, the BLS of Fig. 1 generates random features from the original inputs at the "feature nodes" and broadens the structure at the "enhancement nodes."

In a BLS, the input data are first turned into random features by certain feature mappings, which are then coupled by nonlinear activation

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

functions to construct the enhancement nodes. Then, the output layer is linked to the enhancement layer's and the random features' (nodes') inputs, with the weights of the output layer decided by either a rapid pseudoinverse of the output layer's outputs or a fully connected component. the system equation or a learning procedure based on iterative gradient descent. When there is fresh information to process or the number of augmentation nodes has to grow, incremental learning techniques are used.

When compared to multilayer perceptron and deep structures like CNN, deep belief networks, deep Boltzmann machines, stacked auto encoders, and stacked deep auto encoders, the BLS's efficiency and speed are significantly more advantageous.

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

$$Z_i = \phi_i(XW_{e_i} + \beta_{e_i}), \quad i = 1, 2, \ldots, n \tag{1}$$

where the weights W_{e_i} and bias term β_{e_i} are randomly generated matrices with the proper dimensions.

We denote $Z^n \triangleq [Z_1, Z_2, \ldots, Z_n]$ as the collection of n groups of feature nodes. Then, Z^n is connected into the layer of enhancement nodes.

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

$$H_j \triangleq \xi_j(Z^n W_{h_j} + \beta_{h_j}), \quad j = 1, 2, \ldots, m \tag{2}$$

where ξ_j is a nonlinear activation function. In addition, we denote the outputs of the enhancement layer by $H^m \triangleq [H_1, H_2, \ldots, H_m]$.

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 W_{e_i} is fine-tuned by applying the linear inverse problem (please refer to [3, eq. (4)]).

Therefore, the output Y of a BLS has the following form:

$$Y = [Z_1, Z_2, \ldots, Z_n, H_1, H_2, \ldots, H_m]W^m \tag{3}$$
$$= [Z^n, H^m]W^m$$

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

C. Incremental Learning Algorithms for Broad Learning Systems

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

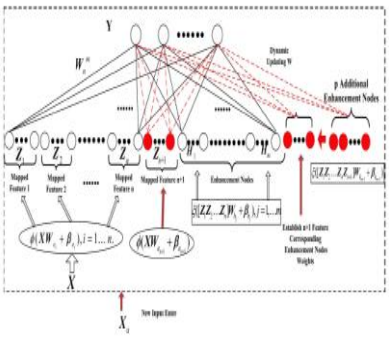


Fig. 2. BLS with increment of input variables, feature nodes, and enhancement nodes (red parts).

It addresses three cases—the growth of enhancement nodes, feature nodes, and input data—in one go (see Fig. 2).

We shall describe the three progressive

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

This is a list of the requirements.

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

$$A^{m+1} \triangleq [A^m, \xi(Z^n W_{h_{m+1}} + \beta_{h_{m+1}})] \tag{4}$$

where $W_{h_{m+1}}$ and $\beta_{h_{m+1}}$ 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

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

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

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

and

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

The preceding formulas show that, instead of computing the pseudoinverse of the whole Am+1, it just has to calculate that of the relevant components, which yields

the BLS capability of rapid learning speed.

Node Feature Increment 2: Inadequate features may be the cause of a discriminative model's (shadow or deep) inability to accurately reflect the input data. Extraction of additional new features by increasing the number of filters or layers and training the models from scratch is a common technique for these architectures, but it may be highly time-consuming, particularly for very deep models.

Even so, adding a new feature mapping to BLS is a breeze to accomplish. Addition of the additional few mapping nodes into the structure is as simple as adding any of the aforementioned improvement nodes, and the connection weights may be educated using the same incremental learning technique.

To illustrate, let's say that the original BLS has n groups of feature nodes and m groups of enhancement nodes, and that we've decided to add an extra n groups of feature nodes, which we'll refer to as n+1.

$$Z_{n+1} = \phi(X W_{e_{n+1}} + \beta_{e_{n+1}}). \tag{8}$$

In addition, the output of corresponding enhancement nodes is

$$H_{ex_m} \triangleq [\xi(Z_{n+1} W_{ex_1} + \beta_{ex_1}), \ldots, \xi(Z_{n+1} W_{ex_m} + \beta_{ex_m})] \tag{9}$$

where W_{ex_i} and β_{ex_i} are randomly generated weights and bias terms that connect the newly added feature nodes to the enhancement nodes.

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

Some online learning situations include a steady stream of training data, meaning we need a model that can easily adjust to the influx of fresh information.

Retraining deep models with all of the original training data is a popular technique. On the other hand, the BLS merely needs to have the weights for the input samples that were just added to the training set updated. Check out the following for further information.

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

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

$$Z_a^n = [\phi(X_a W_{e_1} + \beta_{e_1}), \ldots, \phi(X_a W_{e_n} + \beta_{e_n})] \quad (1$$

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

$$A_x \triangleq \left[Z_a^n, \xi\left(Z_x^n W_{h_1} + \beta_{h_1}\right), \ldots, \xi\left(Z_x^n W_{h_m} + \beta_{h_m}\right)\right]. \quad (1$$

The weights of this incremental BLS can be updated by

$$W_a^m = W^m + \left(Y_a^T - A_x^T W^m\right) B \quad (1$$

where

$$B^T = \begin{cases} C^+, & \text{if } C \neq 0 \\ (1 + D^T D)^{-1} (A^m)^+ D, & \text{if } C = 0 \end{cases}$$
$$C = A_x^T - D^T A^m$$
$$D^T = A_x^T (A^m)^+. \quad (1$$

Similarly, this incremental training process saves time since
it only computes the pseudoinverse of matrix containing the
new part $\boldsymbol{Ax}$. 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 _ $C(\mathbf{I}d)$, which defined on the standard
hypercube $\mathbf{I}d = [0; 1]d$ _ Rd , the BLS with nonconstant
bounded feature mapping _ and activation function _ can equivalently be denoted as

$$f_{w_{m,n}}(x) = \sum_{i=1}^{n*k} w_i \phi(x w_{e_i} + \beta_{e_i}) + \sum_{j=1}^{m*q} w_{nk+j} \xi(z w_{h_j} + \beta_{h_j})$$
$$= \sum_{i=1}^{n*k} w_i \phi(x w_{e_i} + \beta_{e_i}) + \sum_{j=1}^{m*q} w_{nk+j} \xi(x; \{\phi, w_{h_j}, \beta_{h_j}\})$$

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

$$w_{m,n} = (n, m, w_1, \ldots, w_{nk+mq}, w_{e_1}, \ldots, w_{e_{nk}}, w_{h_1}, \ldots, w_{h_{mq}}, \beta_{e_1}, \ldots, \beta_{e_{nk}}, \beta_{h_1}, \ldots, \beta_{h_{mq}})$$

is the set of overall parameters for the functional link network. Among them, the randomly generated part is denoted as $\lambda_{m,n} = (w_{e_1}, \ldots, w_{e_{nk}}, w_{h_1}, \ldots, w_{h_{mq}}, \beta_{e_1}, \ldots, \beta_{e_{nk}}, \beta_{h_1}, \ldots, \beta_{h_{mq}})$. 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 $f_{w_{m,n}}$ and the function f on the compact set $\mathbf{K} \subset \mathbf{I}^d$ can be denoted as

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

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

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

Our main result is as follows.

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

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

Moreover, the randomly generated parameters $\lambda_{m,n}$ are ples from the distributions of probability measures $\mu_{m,}$

Proof: Recall that for function f, the approxir solution of BLS is the function $f_{w_{m,n}}$ defined earlie $w_z = [w_{z_1}, \ldots, w_{z_{nk}}]$ denote the weight matrix conn

the feature nodes Z^n to the output layer, and let w_h = $[w_{h_1}, \ldots, w_{h_{mq}}]$ denote the weight matrix connecting th enhancement nodes H^m to the output layer.

Therefore, for any integer n, define

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

where $w_{e_i}, \beta_{e_i}, i = 1, \ldots, nk$, are samples from the give probability measures. Obviously, the resident function f_{r_n} = $f - f_{w_z}$ is bounded and integrable in $\mathbf{I}^d$ since the featur mapping ϕ is bounded. Furthermore, there exists a functio $f_{c_n} \in C(\mathbf{I}^d)$, 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 guar anteed by the fact in [14]. It is clear that for any $f_{c_n} \in L^2(K)$ 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 tha

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

where $w_{e_i}, \beta_{e_i}, i = 1, \ldots, nk$, are samples from the given probability measures. Obviously, the resident function $f_{r_n} = f - f_{w_z}$ 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 $f_{c_n} \in L^2(K)$, 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

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

where $w_{h_j}, \beta_{h_j}, j = 1, \ldots, mq$, are samples from the given probability measures. Since ϕ and ξ are nonconstant and bounded, the composition function $\xi(x; \{\phi, w_{h_j}, \beta_{h_j}\}), j = 1, \ldots, mq$ 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_{w_h}, such that $\forall \varepsilon > 0$, we have

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

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

Finally, we have that

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

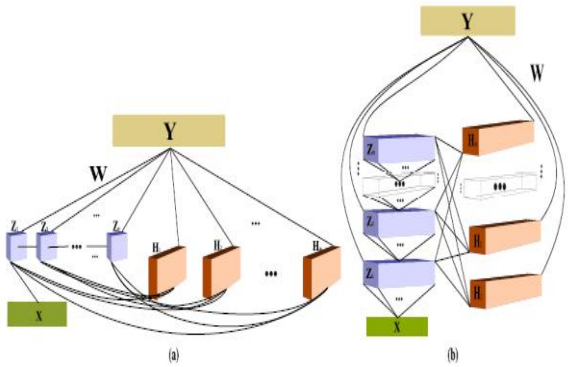
Hence, we could conclude that

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

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 $\{f_{w_{m,n}}\}$ in BLS that is constructed by nonconstant bounded feature mapping ϕ and absolutely integrable activation function ξ (functions on $\mathbf{I}^d$, such that $\int_{\mathbb{R}^d} \xi^2(x)dx < \infty$) and a respective sequence of probability measures $\mu_{m,n}$, such that

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

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



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

The BLS is a nonlinear function approximator, as stated in the aforementioned Theorem, however determining the appropriate weights might be challenging. As we will see,

Various BLS composite models will be discussed. Different connections are established on feature mapping nodes or enhancement nodes in these composite models, which leads to greater nonlinearity mapping for the input and may make it simpler to determine the connection weight in the final layer.

Broad learning system composite models

The BLS may be adjusted to fit a variety of parameters. Particular uses gain greatly from regularization (see [15], [16]). In [17] and [18], a variation is suggested for use in image recognition; it is called graph regularized BLS. To expand upon the original BLS, this section would suggest other frameworks. Many examples and discussions of models are provided. Adopted functions () for building feature nodes and enhancement node functions () have no subscriptions, as is the norm. The following factor often inspires model modifications: First, there's feature map cascading (CFBLS); then, enhancement node cascading (CEBLS); then, limited feature map to enhancement node cascade (LCFBLS); then, limited enhancement map to feature map cascade (LCEBLS); and finally, feature mapping node cascading (CFMCN) and enhancement node cascading (CFMCN) (CFEBLS).

As a whole, learning systems use a 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 $Z_1, Z_2, \ldots, Z_n$ form a cascade connections.

Therefore, for the input data X, the first group of feature mapping nodes Z_1 is denoted as

$$Z_1 = \phi(XW_{e_1} + \beta_{e_1}) \triangleq \phi(X; \{W_{e_1}, \beta_{e_1}\})$$

where W_{e_1} and β_{e_1} are randomly generated by distribution $\rho(w)$. As for the second group, the feature mapping nodes

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

This architecture cascades the a group of feature mapping nodes one after another. As seen in Fig. 3(a), the feature mapping nodes $Z_1, Z_2, \ldots, Z_n$ form a cascade connections.

Therefore, for the input data X, the first group of feature mapping nodes Z_1 is denoted as

$$Z_1 = \phi(XW_{e_1} + \beta_{e_1}) \triangleq \phi(X; \{W_{e_1}, \beta_{e_1}\})$$

where W_{e_1} and β_{e_1} are randomly generated by distribution $\rho(w)$. As for the second group, the feature mapping nodes

0063

Z_2 are established using the output of the Z_1 nodes; therefore, Z_2 is expressed as

$$\begin{aligned} Z_2 &= \phi(Z_1 W_{e_2} + \beta_{e_2}) \\ &= \phi(\phi(XW_{e_1} + \beta_{e_1})W_{e_2} + \beta_{e_2}) \\ &\triangleq \phi^2(X; \{W_{e_i}, \beta_{e_i}\}_{i=1,2}). \end{aligned}$$

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

$$\begin{aligned} Z_k &= \phi(Z_{k-1} W_{e_k} + \beta_{e_k}) \\ &\triangleq \phi^k(X; \{W_{e_i}, \beta_{e_i}\}_{i=1}^{k}), \quad \text{for } k = 1, \ldots, n \end{aligned} \tag{14}$$

where W_{e_i} and β_{e_i} are randomly generated.

Next, the concentrated feature nodes $Z^n \triangleq [Z_1, \ldots, Z_n]$ are connected with the enhancement nodes $\{H_j\}_{j=1}^{m}$, where

$$H_j \triangleq \xi(Z^n W_{h_j} + \beta_{h_j})$$

and W_{h_j} and β_{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:

$$\begin{aligned} Y &= [\phi(X; \{W_{e_1}, \beta_{e_1}\}), \ldots, \phi^n(X; \{W_{e_i}, \beta_{e_i}\}_{i=1}^{n}) \\ &\quad |\xi(Z^n W_{h_1} + \beta_{h_1}), \ \ldots, \xi(Z^n W_{h_m} + \beta_{h_m})]W_n^m \\ &= [Z_1, \ldots, Z_n | H_1, \ldots, H_m]W_n^m \\ &= [Z^n | H^m]W_n^m \end{aligned}$$

where $H^m \triangleq [H_1, \ldots, H_m]$, and W_n^m is calculated through the pseudoinverse of $[Z^n | H^m]$.

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

$$Z_{n+1} \triangleq \phi^{n+1}(X; \{W_{e_i}, \beta_{e_i}\}_{i=1}^{n+1}).$$

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

$$H_{\mathrm{ex}_m} \triangleq [\xi(Z_{n+1} W_{\mathrm{ex}_1} + \beta_{\mathrm{ex}_1}), \ldots, \xi(Z_{n+1} W_{\mathrm{ex}_m} + \beta_{\mathrm{ex}_m})]$$

where $W_{\mathrm{ex}_i}, \beta_{\mathrm{ex}_i}, i = 1, \ldots, m$ are randomly generated.

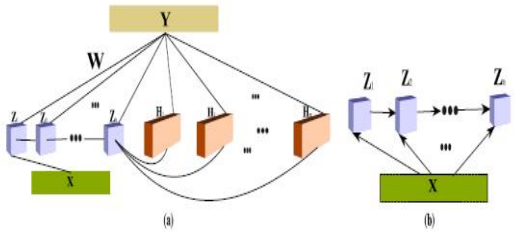


Fig. 4. LCFBLS. Cascade of feature mapping nodes with the last group connected to the enhancement nodes. (a) Broad structure. (b) Alternative feature nodes connection (recurrent structure).

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

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 $A_{n+1}^{m+1} \triangleq [A_n^m|Z_{n+1}|H_{ex_m}|H_{m+1}]$, the updated pseudoinverse and the new weights of this cascade BLS network should be

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

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

where $D = (A_n^m)^+[Z_{n+1}|H_{ex_m}|H_{m+1}]$

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

and $C = [Z_{n+1}|H_{ex_m}|H_{m+1}] - A_n^mD$.

In particular, this network benefits from the speed of
BLS has gradual learning. On the other hand, more distinctive
The updated network is then used to construct features that can
mechanism that can adapt to different situations. Fig. Figure 3(a) is an example of
described in detail above, and Figure 2. Article 3(b) a similar depiction of this BLS cascade network.

B. Extensive Learning Systems' Feature Mapping Cascade

Enhancement Nodes Have Their Last Group Connected
Recurrent Feature Nodes (LCFBLS)

A modified cascaded network is described in Section IV-A. Here,
instead of linking every single node in the feature mapping
points of improvement, the very final set of feature mapping
connection between the improvement nodes and the nodes themselves.
The network with the same output for the same input data X is also
groupings of enhancement nodes (m) and feature node groups (n)

is expressed as follows:

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

where

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

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).

To describe sequential data well, recurrent systems have a concept that is comparable to the cascade of feature mapping [see (14)]. The pattern of repetition

uses temporal information in the input and is thus ideal for text document interpretation and time series processing.
Following [the structure is shown in Fig. 4(b)], the recurrent information may be described in the feature nodes as the recurrent feature nodes. In order to acquire knowledge about successive events

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

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.

On the basis of this version, we can build a recurrent-BLS and a long-term-short-term-BLS. These are the results of the experiments

The suggested models outperform the state-of-the-art approaches in terms of accuracy and training time for a total of 12 natural language processing classification data sets from CrowdFlower, as described in [19].
Nota bene: if the feature nodes are added sequentially, the network topology presented in this section will produce additional improvement nodes. Accordingly, the method is identical to the relevant part of the original BLS in [3], with just the increment of the extra enhancement nodes accessible.
Therefore, in this case, specifics are disregarded.
Cascade of Enhancement Nodes (CEBLS) and Recurrent Enhancement Nodes (REN) are examples of C. General-Purpose Learning Systems.

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

Using a cascade of function composition, the enhancement nodes are reconstructed in the proposed BLS model. Once again, the following equations produce the first n sets of feature nodes from the input data X:

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

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

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

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

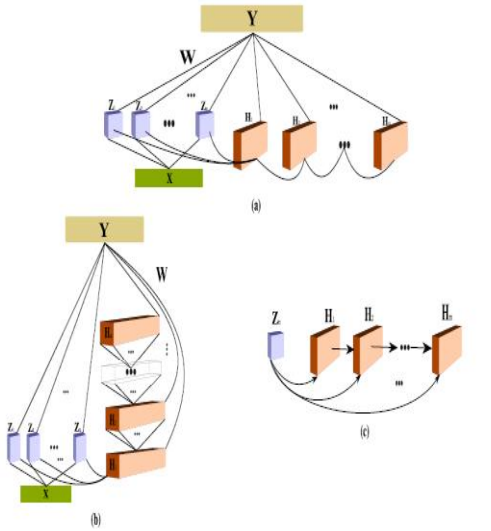


Fig. 5. CEBLS. Cascade of enhancement nodes. (a) Broad structure. (b) Structure redrawn, as right side showing the cascade architecture. (c) Alternative enhancement nodes connection (recurrent structure).

established as follows:

$$\begin{aligned} H_2 &= \xi(H_1 W_{h_2} + \beta_{h_2}) \\ &= \xi(\xi(Z^n W_{h_1} + \beta_{h_1}) W_{h_2} + \beta_{h_2}) \\ &\triangleq \xi^2(Z^n; \{W_{h_i}, \beta_{h_i}\}_{i=1,2}). \end{aligned}$$

Furthermore, the first m groups of enhancement nodes are

$$H_u \triangleq \xi^u(Z^n; \{W_{h_i}, \beta_{h_i}\}_{i=1}^u), \quad \text{for } u = 1, \ldots, m \quad (18$$

where W_{h_i} and β_{h_i} are randomly generated under the given distribution.

Consequently, the nodes Z^n and $H^m \equiv [H_1, \ldots, H_m]$ are connected directly with the output, and the modified BLS is

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

and W_n^m is calculated through the pseudoinverse of $[Z^n | H^m]$

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 $Z_{n+1} \triangleq \phi(XW_{e_{n+1}} + \beta_{e_{n+1}})$. The uth group of enhancement nodes should be supplemented by $\xi^u(Z_{n+1}; \{W_{\text{ex}_i}, \beta_{\text{ex}_i}\}_{i=1}^u), u = 1, \ldots, m$ and the corresponding matrix is denoted as

$$H_{\text{ex}_m} \triangleq \left[\xi(Z_{n+1}; \{W_{\text{ex}_1}, \beta_{\text{ex}_1}\}), \ldots, \xi^m\left(Z_{n+1}; \{W_{\text{ex}_i}, \beta_{\text{ex}_i}\}_{i=1}^m\right)\right]$$

where $W_{\text{ex}_i}, \beta_{\text{ex}_i}, i = 1, \ldots, m$ are randomly generated.

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

$$H_{m+1} \triangleq \xi^{m+1}\left(Z^{n+1}; \{W_{h_i}, \beta_{h_i}\}_{i=1}^{m+1}\right)$$

where $Z^{n+1} \triangleq [Z_1, \ldots, Z_{n+1}]$, and $W_{h_{m+1}}$ and $\beta_{h_{m+1}}$ are randomly sampled. Therefore, the matrix $A_n^m \triangleq [Z^n | H^m]$ is updated as $A_{n+1}^{m+1} \triangleq [A_n^m | Z_{n+1} | H_{\text{ex}_m} | H_{m+1}]$. In fact,

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

the output weights W_{n+1}^{m+1} could be dynamically updated under (15)–(17) since the notations of A_n^m and 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

$$H_j = \xi(H_{j-1}W_{h_j} + Z^n W_{z_j} + \beta_{h_j}), \quad j = 1, \ldots, m$$

where W_{z_j} is the added weights for the features Z^n. To test the experimental results on two generally chaotic systems are presented in [20] to assess the performance of the variations in time series. For the provided benchmark datasets, prediction accuracy is noticeably

superior in performance over previous models

D. Generalized Learning Systems, with a Hierarchy of Feature Mapping and Boosting Nodes (CFEBLS)

Here, we use a whole cascade of nodes for mapping features and nodes for boosting those maps. The composite feature nodes Zk, k = 1,..., n, are produced once again for a given input data set X and output data Y.

$$Z_k \triangleq \phi^k\left(X; \{W_{e_i}, \beta_{e_i}\}_{i=1}^k\right)$$

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

$$H_u \triangleq \xi^u\left(Z^n; \{W_{h_i}, \beta_{e_i}\}_{i=1}^u\right), \quad \text{for } u = 1, \ldots, m$$

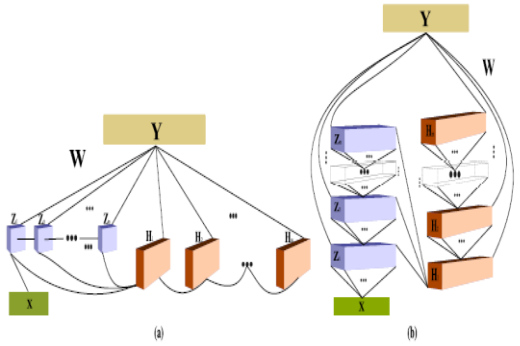


Fig. 6. CFEBLS. Comprehensive cascade composition model. (a) Broad structure. (b) Structure redrawn.

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

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 $A_{n+1}^{m+1} \triangleq [A_n^m | Z_{n+1} | H_{ex_m} | H_{m+1}]$, where

$$Z_{n+1} \triangleq \phi^{n+1}\left(X; \{W_{e_i}, \beta_{e_i}\}_{i=1}^{n+1}\right)$$
$$H_{ex_m} \triangleq \left[\xi(Z_{n+1}; \{W_{ex1}, \beta_{ex1}\}), \ldots, \xi^m\left(Z_{n+1}; \{W_{ex_i}, \beta_{ex_i}\}_{i=1}^{m}\right)\right]$$
$$H_{m+1} \triangleq \xi^{m+1}\left(Z^{n+1}; \{W_{h_i}, \beta_{h_i}\}_{i=1}^{m+1}\right)$$
$$Z^{n+1} \triangleq [Z_1, \ldots, Z_{n+1}]$$
$$H^{m+1} \triangleq [H_1, \ldots, H_{m+1}]$$

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).

Note that an alternate BLS architecture exists in which each cascade group of feature mapping nodes is linked to each cascade node in the enhancement cascade. Nevertheless, the equations

the planned ones, and specifics are not included here.

E. General-Purpose Learning Systems: The Hybrid Model vs. Distributed and Embedded Learning

A single-layer linear system and deep neural networks are combined in the broad and deep learning architecture recently proposed ([21], [22]). See Fig. 5(a) and (b) in Section IV-C to see how this model compares to our own construction.

The cascade models suggested in this research may be rebuilt in a deep structure that is equal to the original BLS's flattened neural network architecture.

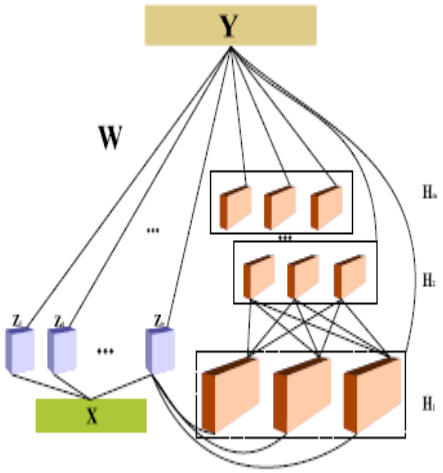


Fig. 7. CEBLS. Modified network for cascade of enhancement nodes.

Specifically, 3b, 5b, and 6b in Figures 3, 5, and 6. This revised set of models is not just "wide," but also "deep."

To recap, the first BLS included n sets of feature maps.

Figure 1 depicts a network with n nodes for enhancement and m clusters. The updated system is similar to the model in [21], with the exception of the full-link connection between the m + n groups of nodes in BLS and the output layer, if the weights connecting the first n1 groups of feature maps and the first group of enhancement nodes are mandated to be 0. With these adjustments, the original network is shown in Fig. 7.

F. Distributed Learning Architectures with a Feature Mapping Node Cascade Generated by Convolutions (CCFBLS)

The convolutional neural network (CNN) has shown to be a useful tool for pattern identification provided the weights between the layers are properly set. For the sake of this discussion, the model shown in Fig. 8 may be thought of as a BLS model with a cascade of convolution functions, with the nodes representing feature mappings that are being mapped using convolution and pooling operators.

In other words, CFBLS (discussed in Section IV-A) is a cascade of convolution feature mapping nodes, of which this model is a special instance (CCFBLS).

Under the cascade of convolution and pooling operations in the feature mapping nodes, the network based on convolutional functions is built. In the first step, we specify the feature mapping nodes () as follows:

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

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

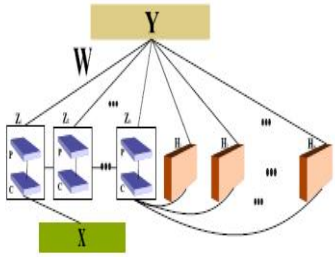


Fig. 8. CCFBLS. Cascade of Convolution feature mapping nodes.

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

where the operator $\otimes$ is the convolutional function for tl given matrices, the function $P(\cdot)$ is the pooling operatio and $\theta(\cdot)$ is the selected activation function. Furthermor the weights of convolutional filters are randomly sample under a given distribution. Second, the expected netwo is enhanced by function $H_j \triangleq \xi(Z^n W_{h_j} + \beta_{h_j})$, for j $1, \ldots, m$, where $Z^n \triangleq [Z_1, \ldots, Z_n]$. Finally, to ensure much as information passed into the output layer, Z^n and H are linked to the intended Y in a straightforward way. Figure 8 depicts the whole network. Connections in this design are created in a manner similar to that of a 3-D convolutional neural network (CNN).

layers deep down to the last layer where the output is generated. Data from CIFAR-10 and CIFAR-100 have been used to evaluate this combined model. Compared to previous versions, this one is more faster and more accurate [23].

G. Feature-Node Fuzzy Modeling for a Fuzzy-Theoretical Broad Learning System

Fuzzy BLSs may be created by combining BLS with the Takagi-Sugeno (TS) fuzzy scheme. Instead of using traditional BLS feature nodes, the fuzzy version of this algorithm uses a collection of TS fuzzy subsystems.

In order to maintain input characteristics, the outputs of fuzzy rules generated by all fuzzy subsystems in the feature nodes are transferred to the enhancement layer for further nonlinear processing. Refer to [24] for further information on fuzzy BLS.

Experiments, Part V

In Section IV, we analyzed and proposed many BLS variations, some of which include a cascade of feature mapping and/or enhancement nodes. Additional applications are built and submitted in [19], [20], and [24], and these frameworks have been tested on various data sets. Here, we compare many data sets using both the standard BLS and several of its modifications.

The original BLS is utilized in the following tests with other popular models including support vector machines, linear support vector machines, and extreme learning machines on representative benchmarks for function approximation, time series prediction, and facial recognition.

Appropriate Approximation of Functions

UCI Regression Data Sets (Set 1): From the database at the University of California, Irvine (UCI) [25], we choose ten sets of regression data that span three sizes and dimensionalities: small, medium, and big. Table I contains information about the available data.

There is a significant impact from the cost parameter C and kernel parameter of SVM, LSSVM [26], and ELM [27].

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

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

The root-mean-squared errors (RMSE) of SVM, LSSVM, ELM, and BLS are shown in Table III,

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

and we choose the best results from 10 trials for each data set.

We may draw the conclusion that across all 10 function approximation data sets, the BLS is the most accurate testing method, followed by the SVM, LSSVM, and ELM.

Second, Time-Series Forecasting

Using a dataset of wind speeds [29], we evaluate the accuracy of BLS, AR, an adaptive network-based fuzzy inference system (ANFIS), support vector machines (SVMs), and predictive deep Boltzmann machines (PDBMs) in making predictions about future wind speeds. Wind speed data is gathered at a rate of 50,000 every 10 minutes for use in training and 2500 per 10 minutes for use in testing.

In the experiment, we utilize the average of the last ten wind speeds as input for the forecast. All 50,000 samples are used to train the models, and then the models are used to provide predictions about wind speed for the next 10 minutes to two hours based on testing data. It is decided to use the MAPE to compare the models. The criteria 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
Abalone	2^2	2^{-1}	2.8932	3.0774	2^0	2^0	5	6
Basketball	2^0	2^0	6.0001	27.3089	2^{25}	2^{11}	6	7
Bodyfat	2^2	2^{-2}	6505.7167	233.8448	2^{14}	2^5	6	5
Cleveland	2^2	2^2	0.7527	45.2507	2^{13}	2^{15}	1	10
Housing	2^2	2^1	61.9215	6.4770	2^6	2^0	5	29
Mortgage	2^{10}	2^{-1}	803.9607	5.4985	2^{13}	2^3	9	4
Pyrim	2^{10}	2^8	52.5877	3.2463	2^2	2^6	3	7
Quake	2^5	2^5	0.1115	0.0079	2^5	2^{14}	10	2
Strike	2^0	2^{-4}	0.3167	0.7383	2^{-1}	2^5	9	11
Weather Izmir	2^4	2^{-2}	1433.0467	44.2146	2^{12}	2^2	4	3

TABLE III

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

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

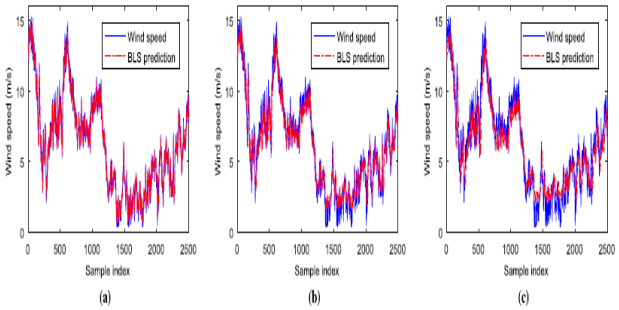


Fig. 9. Wind speed prediction of BLS for (a) 10 min ahead, (b) 60 min ahead, and (c) 120 min ahead.

Figure 9 displays the outcomes of BLS prediction at 10, 60, and 120 minutes in advance. The BLS's AE is the smallest, and we can see that AEs from other models react in a variety of ways, but they all follow a general trend.

extreme amplitude of oscillations.

Table IV compares the models' performances; it's clear that the BLS provides the most accurate prediction of short-term wind speeds.

Recognizing People's Faces, Part C

This section compares BLS's classification efficacy to that of SVM, LSSVM, and ELM using three widely-used face data sets: Extend YaleB [30], The ORL Database of Faces (ORL) [31], and UMIST [32]. Below you'll find more information on the aforementioned face data sets.

For starters, there's the 32x32px-sized cropped photos of 38 participants that make up the enlarged YaleB face database. There is a wide range of lighting and facial expressions in these pictures.

Each individual is represented by 30 training photographs and the remaining 1274 testing images.

2. ORL: The ORL data collection contains 400 grayscale facial photos of 40 unique people, each with a dimension of 32x32, captured between April 1992 and April 1994 at AT&T Laboratories Cambridge.

Six photographs are selected at random from each person's collection, and the remaining 160 are used for testing.

Third, UMIST; the UMIST face database has 575 photos of 20 unique participants at a resolution of 112 x 92.

The dataset is more difficult than average because to the bigger variances between photographs of the same face in viewing direction than ordinary image variations in face identity. We next resize them to 56 by 46 pixels and choose 15 photographs at random for training and the remaining images for testing for each participant.

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

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

Min. ahead	MAPE×10^{-2}				
	AR	ANFIS	SVM	PDBM	B
10	7.57	10.84	14.55	7.05	6
20	9.68	14.96	17.01	8.80	8
30	11.19	18.01	19.04	10.12	9
40	12.55	20.49	21.03	11.40	11
50	13.71	22.20	22.19	12.25	11
60	14.78	23.60	23.35	12.98	12
70	15.74	25.12	24.96	13.69	13
80	16.58	26.38	25.93	14.16	13
90	17.41	27.61	27.05	14.81	14
100	18.15	28.64	27,29	15.48	15
110	18.95	29.69	27.32	16.01	15
120	19.73	30.48	28.64	16.63	16

Grid search is used to determine the best parameters for these models, using the same searching intervals used for support vector machines and extreme learning machines.

Extending the searching parameters for BLS to [1, 60][1, 50] [1, 6000] is possible. Table V displays the parameter settings, while Table VI displays the categorization outcomes.

Across all three datasets, the BLS consistently beats the widely used discriminative models at recognizing faces.

D. Variant BLS Categorization

Here we examine the classification efficacy of BLS and its variations using the MNIST data set [33] and the NORB data set [4].

1) MNIST: The MNIST dataset contains 60,000 training pictures and 10,000 testing images across 10 classes, all of which are 28 by 28 pixels in size. Meanwhile, the incremental algorithms of the variations are not presented in this part; that is, only the one-shot versions are studied and compared due to the length constraint of this study.

Second, NORB, whose data set includes 48,600 pictures with resolutions of 2, 32, 32 pixels apiece. Animals, people, aircraft, trucks, and automobiles are the five types of vehicles included in the dataset, respectively. Two sets of photos, each including 24 300 pictures, are chosen: one set is used for training, while the other set is used for testing.

The cascade of the feature maps and the enhancement nodes is always set to 2 since it is not difficult to classify MNIST data and NORB data. To find these parameters, we do a grid search that includes the following categories: two sets of numbers: 1) the numbers of two-layer-cascaded feature nodes N f c, the mapping groups Nmc, and the enhancement nodes Ne for the cascade of the feature mapping nodes (CFBLS); and 2) the numbers of two-layer-cascaded feature nodes Nflc, the mapping groups Nmlc, and the enhancement nodes Ne. Ne for the restricted linkage between cascaded feature map groups and enhancement nodes (LCFBLS); 3) the feature node, mapping group, and enhancement node counts (N f, N m, and Nec, for two-layer cascaded enhancement nodes); number of mapping groups (Nmlc), number of enhancement nodes (CEBLS), number of two-layer-cascaded feature nodes (Nflc), and number of two-layer-cascaded enhancement nodes (Nflc).

Node-Enhancement-Mapping-Cascade (Nelc) for Feature Mapping (CFEBLS). For the sake of brevity, we'll refer to the number of feature nodes as N f, the number of mapping groups as N m, and the number of enhancement nodes as Ne.

Tables VII and VIII detail the MNIST and NORB categorization outcomes, respectively. The accuracy measurements on the MNIST and NORB benchmarks are clearly comparable. The positive qualities it has may be to blame for this.

The networks all have somewhat different architectures, which suggests that the cascade variations perform better than the default BLS in terms of structure optimization and, in most cases, fewer parameters. There are other applications produced and filed in [19], [20], and [24].

E. A Study of Resnet-34 and CCFBLS for Face Recognition

It is demonstrated in Section IV-F and Figure 8 that MS-Celeb-1M [34], one of the most difficult large-scale face recognition databases, is utilized to evaluate the BLS composite model employing convolution feature nodes, CCFBLS. This dataset was created as a benchmark task for facial recognition with the goals of 1) identifying one million celebrities from their face images and linking them to the appropriate entity keys in a knowledge base [34] and 2) exploring low-shot face recognition with the end goal of developing a large-scale face recognizer capable of recognizing a large number of individuals with high precision and high recall [35].

Figure 10 shows a selection of photos taken from the MS-Celeb-1M datasets.

The following are the planned procedures for the experiment. Each individual in the initial data set had between fifty and one hundred photos. There are 20,000 people in the basic set and 1,000 people in the novel set. When it comes to training the face representation model, the base set provides tens of

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

photographs for each celebrity, while the novel set only provides a single image. For the purpose of testing the proposed CCFBLS, we only used the first 2000 human photos from the base set in our experiment.

MS-Celeb-1M was divided into a training set of 119,134 color pictures linked with 2000 people and a testing set of 10,000 color images. On average, photo dimensions are 2503003. When compared to the picture sizes used in the Extended YaleB, ORL, and UMIST databases, which are all 32x32 pixels, this is a very huge size, and it undoubtedly enhances the complexity and difficulty of the learning.

A comparison is made between the results of the proposed CCFBLS and those of the residual network, which has shown to be a potent and widely used tool in image processing and recognition. A regular 34-layer residual network (Resnet-34) [36] is built, with no additional features or gimmicks. There are just 18 convolution functions used to build the CCFBLS, and only four of those convolution outputs are linked to the CCFBLS's output nodes.

Both simulations run on a PC outfitted with an Intel Corei7-7800X and an 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 exan samples are selected from base set [35].

The trials were carried out in an Ubuntu Linux setup, utilizing Tensorflow version 1.7.0.

Table IX displays all the results in tabular form. The sum of all

However, Resnet-34 only has a total of 188 880 neurons, when the required number is 108 196. CCFBLS employs a total of 12.290 million parameters, while 23.794 million

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

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%**

Resnet-34 makes use of a set of parameters. It's shown that the suggested CCFBLS requires around 50% less neurons, parameters, and training time to get a superior testing performance.

accuracy.

A Final Thought

We offer many alternative BLS architectures and describe their essential features. The goal of setting up such a facility is to provide researchers with options for building flattened networks. In these versions, weight connections may be made either inside the feature mapping nodes, within the enhancement nodes, or between the feature mapping nodes and the enhancement nodes, but the incremental learning techniques from the original BLS can still be used.

These versions are also modeled mathematically for your perusal. The suggested BLS versions may be thought of as a specific arrangement of several deep and broad neural networks [10, [12], [13].

To further demonstrate the usefulness of BLS, we use results by Hornik to show that any measurable function on Rd may be arbitrarily well approximated by a BLS with nonconstant bounded feature mapping and activation function in measure.

The regression performance of BLS is compared to that of SVM, LSSVM, and ELM on the UCI database and facial recognition data sets, such as Extend YaleB, ORL, and UMIST, in order to evaluate its approximation capabilities. Additionally, BLS's time series prediction is compared with those of AR, ANFIS, SVM, and PDBM. Parameters that allow for the highest possible testing accuracy are developed for each method using a grid search so that they may be fairly compared. The categorization skills of BLS variations are then put to the test in NORB and MNIST.

Cascade Convolution Feature Mapping Nodes BLS (CCFBLS) is a variation of BLS that has been proven to improve testing recognition accuracy while using almost half the normal number of feature mapping nodes.

compares Resnet-34 in terms of neurons, parameters, and training time on the MS-Celeb-1M large-scale picture data set. It is shown, using the standard statistics, that the BLS

and its versions fare better on tests than the aforementioned algorithms.

REFERENCES

According to [1] "A rapid learning method for deep belief networks" by G. E. Hinton, S. Osindero, and Y.-W. Teh, published in Neural Comput., vol. 18, no. 7, pp. 1527-1554, 2006.

[2] "Handwritten digit recognition using a back-propagation network," by Y. LeCun and colleagues in Proc. Adv. Neural Inf. Process. Syst., 1990, pp. 396-404.

IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 1, pp. 10-24, Jan. 2018, doi: 10.1109/TNNLS.2017.2716952, which cites "Broad learning system: an effective and efficient incremental learning system without the requirement for deep architecture" by C. L. P. Chen and Z. L. Liu.

[4] "Learning approaches for generic object identification with invariance to position and illumination," by Y. LeCun, F. J. Huang, and L. Bottou, published in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), vol. 2, pp. 97-104, June 2004.

According to [5] "Characteristics of the functional link net: A higher order delta rule net," written by M. Klassen, Y. Pao, and V. Chen and published in Proc. IEEE Int. Conf. Neural Netw. in July 1988, pages 507-513.

Int. J. Control, vol. 56, no. 2, pp. 263-289, 1992; Y.-H. Pao, S. M. Phillips, and D. J. Sobajic, "Neural-net computing and the intelligent control of systems."

"Multilayer feedforward networks are universal approximators," Neural Netw., volume 2, issue 5, pages 359-366, 1989, K. Hornik, M. Stinchcombe, and H. White.

"Approximation capabilities of multilayer feedforward networks," by K. Hornik, was published in Neural Netw., volume 4, issue 2, pages 251-257, 1991.

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

For example, see [10] "A thorough overview on functional link neural networks and an adaptive PSO-BP learning for CFLNN," by S. Dehuri and S.-B. Cho in Neural Comput. Appl., volume 19, issue 2, pages 187-205.

A fast supervised learning neural network for function interpolation and approximation. C. L. P. Chen. IEEE Trans. Neural Netw., vol. 7, no. 5, pp. 1220-1230, September 1996.

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

An Incremental Adaptive Implementation of Functional-Link Processing for Function Approximation, Time-Series Prediction, and System Identification by C. L. P. Chen, S. R. LeClair, and Y.-H. Pao, Neurocomputing, Vol. 18, Nos. 1-3, Pages 11-31, 1998.
(13), "A quick learning and dynamic stepwise updating technique for flat neural networks and the application to timeseries prediction," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 29, no. 1, pp. 62-72, Feb. 1999.
Specifically: [14] W. Rudin, Real and Complex Analysis (Higher Mathematics Series).
McGraw-Hill, 1987. New York, NY, USA.
When it comes to estimating the parameters of regularization in feedforward neural networks, [15] P. Guo, M. R. Lyu, and C. L. P. Chen write, Article first published in IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Volume 33, Issue 1, Pages 35–44, February 2003.
Xu, X. Chang, F. Xu, and H. Zhang, "l1/2regularization: A thresholding representation theory and a rapid solution," [16] Those interested may find the whole article in IEEE Transactions on Neural Networks and Learning Systems, Issue 7 (July 2012), Pages 1013-1027.
Discriminative graph regularized wide learning system for image identification, J. W. Jin, Z. L. Liu, and C. L. P. Chen, to appear in Sci. China Inf. Sci., doi: 10.1007/s11432-017-9421-3.
Neurocomputing, forthcoming, "Regularized robust wide learning system for uncertain data modeling" [18] by J. W. Jin, C. L. P. Chen, and Y. T. Li.
"Accurate and efficent text categorization by simultaneous learning of different information utilizing recurrent and gated wide learning system," will be published in IEEE Trans. Cybernatics [19] by C. M. Vong, J. Du, and C. L. P. Chen.
According to [20] "Recurrent wide learning systems for time series prediction," by M. L. Xu, M. Han, C. L. P. Chen, and T. Qiu, which will appear in IEEE Trans. Cybern.
In Proc. 1st Workshop Deep Learn. Recommender Syst., 2016, pp. 7-10, H.-T. Cheng et al. present "Wide & deep learning for recommender systems."
"To go deep or broad in learning?" by G. Pandey and A. Dukkipati in Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS), Reykjavik, Iceland, 2014. Pages 724-732.
Q. Y. Feng, Z. L. Liu, and C. L. P. Chen, "Composite convolution kernels in wide learning systems for image recognition," to appear in IEEE Trans. Cybernatics [23].

As an example, see [24] "Fuzzy wide learning system: A new neurofuzzy model for regression and classification," to be published in IEEE Trans. Cybern., doi: 10.1109/TCYB.2018.2857815.
UCI collection of machine learning datasets, C. L. Blake and C. J. Merz, vol. 55, Dept. Inf. Comput. Sci., University of California, Irvine, Irvine, CA, USA, 1998. [Online]. The datasets may be found at: http://archive.ics.uci.edu/ml/ datasets.html.
[26] Least Squares Support Vector Machines, by B. De Moor and J. Vandewalle.
Publisher: Singapore: World Scientific, 2002.
[27] Extreme learning machine for regression and multiclass classification, G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 42, no. 2, pp. 513-529, Apr. 2012.
According to [28] "LIBSVM: A library for support vector machines," by C.-C. Chang and C.-J. Lin, published in ACM Trans. Intell. Syst. Technol., volume 2, issue 3, 2011, article no. 27.
"Predictive deep Boltzmann machine for multiperiod wind speed forecasting," C.-Y. Zhang, C. L. P. Chen, M. Gan, and L. Chen, IEEE Trans. Sustain. Energy, volume 6, issue 4, pages 1416-1425, October 2015.
According to [30] "Acquiring linear subspaces for face recognition in varied lighting," K.-C. Lee, J. Ho, and D. Kriegman, IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 5, pp. 684-698, May 2005.
This is according to [31] "Parameterisation of a stochastic model for human face recognition," by F. S. Samaria and A. C. Harter, published in Proc. 2nd IEEE Workshop Appl. Comput. Vis., December 1994, pages 138-142.
Characterizing Virtual Eigensignatures for Broad-Scale Face Recognition, in Face Recognition, by D. B. Graham and N. M. Allinson (32]. Springer, 1998, pp. 446–456. Berlin, Germany.
Gradient-based learning applied to document recognition, Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998 [33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner.
MS-Celeb-1M: A dataset and benchmark for large-scale face recognition. Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016.
Both Y. Guo and L. Zhang are credited in [35]. (2017). Instantaneous identity verification via the uplift of traditionally marginalized groups. [Online]. You may find the paper at https://arxiv.org/ abs/1707.05574.
Reference: [36] "Deep residual learning for picture recognition," by K. He, X. Zhang, S. Ren, and J. Sun. [Online]. This paper may be found at https://arxiv.org/ abs/1512.03385.

International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-11 Issue-02 Dec 2022

C. L. Philip Chen (S'88, M'88, SM'94, F'07) earned a doctorate from the University of Michigan in Ann Arbor.

Currently he serves as the Chair Professor at the Department of Computer and Information Science in the Faculty of Science and Technology at the University of Macau in Macau, China. His greatest accomplishment is getting the University of Macau's Engineering and Computer Science programs accredited by the Washington/Seoul Accord through the Hong Kong Institution of Engineers (HKIE), Hong Kong; he is a Program Evaluator for the Accreditation Board for Engineering and Technology Education (ABET), Baltimore, MD, USA, for computer engineering, electrical engineering, and software engineering. His current work focuses on the intersection of systems, cybernetics, and AI.
Dr. Chen is an active member of the scientific community as a fellow of many prestigious organizations, including the AAAS, the IAPR, the CAA, and the HKIE. His alma school, Purdue University, honored him by bestowing upon him the Outstanding Electrical and Computer Engineers Award for 2016. From 2015 to 2017, he presided over the International Federation of Automatic Control's Technical Committee 9.1, which focuses on economic and business systems. He serves as the Chief Editor of the IEEE Transactions on Systems, Man, and Cybernetics: Systems, and as an Associate Editor for the IEEE Transactions on Fuzzy Systems and Cybernetics. He served as president of the IEEE Systems, Man, and Cybernetics Society from 2012 to 2013. In his present role, he serves as CAA's vice president.

Bachelor's and Master's degrees in mathematics were earned by Zhulin Liu from Shandong University in Jinan, Shandong, China in 2009 and 2012, respectively.

He is presently enrolled in the Ph.D. program at the Faculty of Science and Technology at the University of Macau (Macau, China).
She is now exploring topics in artificial intelligence, pattern recognition, and function approximation.

IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS

Beijing Normal University in Beijing, China, awarded Shuang Feng a Bachelor of Science in Mathematics and a Master of Science in Applied Mathematics.

between the years of 2005 and 2008 Right now he is a doctoral student in computer science at the University of Macau in Macau, China.
He is an Associate Professor at Beijing Normal University in Zhuhai, China, where he teaches applied mathematics. His current work focuses on computational intelligence applications of fuzzy systems and fuzzy neural networks.