**Article:**
Zhang, J, Li, Y, Xiao, W et al. (1 more author) (2020) Non-iterative and Fast Deep Learning: Multilayer Extreme Learning Machines. Journal of the Franklin Institute, 357 (13). pp. 8925-8955. ISSN 0016-0032

https://doi.org/10.1016/j.jfranklin.2020.04.033

# Non-iterative and Fast Deep Learning: Multilayer Extreme Learning Machines

Jie Zhang[a], Yanjiao Li[b], Wendong Xiao[c,*], Zhiqiang Zhang[d]

[a]*School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China*
[b]*School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China*
[c]*School of Automation & Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[d]*School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, UK*

**Abstract**

In the past decade, deep learning techniques have powered many aspects of our daily life, and drawn ever-increasing research interests. However, conventional deep learning approaches, such as deep belief network (DBN), restricted Boltzmann machine (RBM), and convolutional neural network (CNN), suffer from time-consuming training process due to fine-tuning of a large number of parameters and the complicated hierarchical structure. Furthermore, the above complication makes it difficult to theoretically analyze and prove the universal approximation of those conventional deep learning approaches. In order to tackle the issues, multilayer extreme learning machines (ML-ELM) were proposed, which accelerate the development of deep learning. Compared with conventional deep learning, ML-ELMs are non-iterative and fast due to the random feature mapping mechanism. In this paper, we perform a thorough review on the development of ML-ELMs, including stacked ELM autoencoder (ELM-AE), residual ELM, and local receptive field based ELM (ELM-LRF), as well as address their applications. In addition, we also discuss the connection between random neural networks and conventional deep learning.

*Keywords:* Deep Learning, Multilayer Extreme Learning Machine, Stacked Extreme Learning Machine Autoencoder, Residual Extreme Learning Machine, Local Receptive Field based Extreme Learning Machine

## 1. Introduction

In the past few years, machine learning has been intensively studied both in theory and applications, and is powering many aspects of our daily life. Current machine learning techniques can deal with some kinds of tasks more efficiently than human beings, but their cognitive capability, flexibility, and robustness are still weak. Among all the machine learning techniques, neural network is one of the most important branches, which can theoretically model any "black-box" process. Generally, the development of neural network arises from the following two objectives: 1) understand the nervous system of human brain better; and 2) build information processing system to make data-driven decision and prediction based on biological mechanism [1].

Conventional machine learning techniques usually cannot function well in dealing with complex tasks with high-dimension data. In order to build a satisfactory machine learning model, artificial feature extraction is usually designed for achieving more discriminative features, but only for specific tasks, which is time-consuming and labor-intensive. In addition, feature extraction and machine learning modeling are usually two isolated independent phases without intrinsic connections, which may cause that the created machine learning models have no awareness of the noise existed in the data and degrade their performance. Accordingly, representation learning is important, which can automatically extract more representative features from the raw data. Deep learning approaches are actually with multiple levels of representation, whose hierarchical structure can learn from data using a general-purpose learning procedure without specific design for corresponding tasks [2]. In 2006, the layer-wise-greedy learning marked the birth of deep learning, whose basic idea is that unsupervised learning is implemented for network pre-training, sequentially layer-by-layer

*Corresponding author
*Email address:* wdxiao@ustb.edu.cn (Wendong Xiao)

learning for automatic feature extraction, and the whole deep neural network will be fine-tuned finally. It means that deep neural networks consist of a hierarchical structure with several layers, each of which is actually a non-linear information processing unit. This kind of hierarchical structure can guarantee the capability of deep neural networks for representing complex target functions if the number of layers or units is increased gradually. The past decade witnesses the breakthroughs of deep learning in image recognition, computer vision, and natural language process, etc., motivating many researchers to study feedforward neural networks with hierarchical structure [3]. However, conventional deep learning approaches, including deep belief network (DBN), restricted Boltzmann machine (RBM), and convolutional neural network (CNN), etc. [4, 5, 6], suffer from time-consuming training process due to fine-tuning of a large number of parameters and the complicated hierarchical structure. In addition, the above complication makes it difficult to theoretically analyze and prove the universal approximation of those deep learning approaches. Differently, this paper focuses on a special way for constructing deep neural networks based on extreme learning machine (ELM) theory.

Classic ELM is a type of generalized single hidden layer feedforward networks (SLFNs) [7]. Different from the traditional gradient-based training approaches for SLFNs, which are easy to trap in the local minimum and time-consuming, the hidden layer parameters of ELM are assigned randomly, and the output weights are then analytically calculated through the least-square method [8]. ELM training involves two phases, including randomly generation of hidden layer parameters from a predefined specific interval, and calculation of generalized inverse of the output weight matrix. Thus, ELM is much faster and easier to implement than most state-of-the-art machine learning approaches. In the past decade, ELM theory and applications have attracted numerous attention, its variants and extensions have been developed for specific problems, such as online sequential learning [9], imbalance learning [10], multilabel learning [11], compressive learning [12], and compact modeling [13], etc. ELM was recently extended to hierarchical structure for dealing with complex tasks, i.e., multilayer ELM (ML-ELM). Experimentally, compared with conventional deep learning approaches, ML-ELM shows comparative performance, but much faster learning speed.

Considering ML-ELM may accelerate the development of deep learning, this paper makes a comprehensive survey on ELM-based deep learning, especially discusses the differences between ELM-based deep learning approaches and conventional deep learning approaches with our comments and remarks. It should note that, although there are some surveys about ELM [14, 15, 16, 17], it rarely involves ML-ELM. Thus, regarding the focuses, this paper is quite different from other existing related surveys.

The paper is organized as follows: ELM theory is firstly introduced in Section 2. Stacked ELM autoencoder (ELM-AE), residual ELM, and local receptive field based ELM (ELM-LRF) are then reviewed in Section 3, Section 4, and Section 5, respectively. Selected applications are highlighted in Section 6. Finally, discussions are given in Section 7, and followed by conclusions in Section 8.

## 2. ELM Theory

In this section, ELM theory is briefly introduced to facilitate the understanding of the following sections.

As illustrated in Fig. 1, classic ELM consists of input layer, hidden layer, and output layer. ELM was originally proposed for the SLFNs and was extended to the generalized SLFNs where the hidden layer need not be neuron alike. Different from the existed machine learning approaches, its hidden layer parameters are generated randomly, so the learning problem is transformed as the estimation of the optimal output weights $\boldsymbol{\beta}$ for a given dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\tilde{N}} \subset R^n \times R^m$. ELM training includes two phases: 1) random feature mapping, mainly for feature mapping from the original input space to the ELM feature space; 2) linear parameters solving, mainly for calculating the output weights.

Generally, ELM can be treated as a linear combination of $L$ activation functions:

$$f(\mathbf{x}) = \sum_{i=1}^{L} \beta_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{1}$$

where $L$ represents the number of hidden nodes of ELM, and $\mathbf{h}(\mathbf{x})$ represents the mapped feature vector.

The corresponding matrix form can be expressed as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y} \tag{2}$$

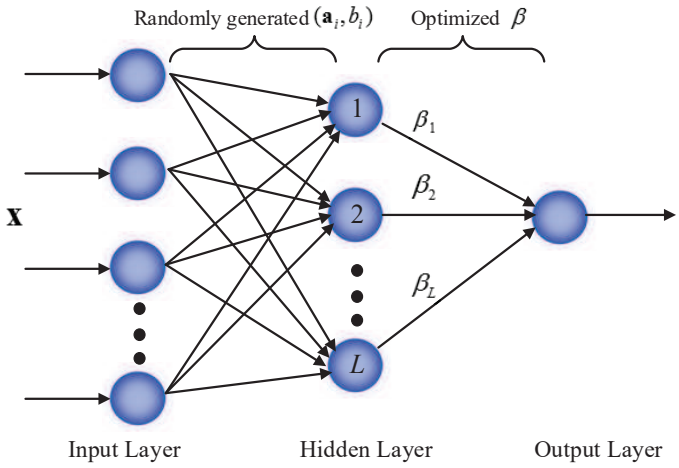



Figure 1: Structure of Classic ELM

where $\mathbf{H}$ represents the hidden layer output matrix:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_{\tilde{N}}) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_{\tilde{N}}) & \cdots & h_L(\mathbf{x}_{\tilde{N}}) \end{bmatrix} \tag{3}$$

and $\mathbf{Y}$ represents the training data target matrix:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_{\tilde{N}}^T \end{bmatrix} = \begin{bmatrix} y_{11} & \cdots & y_{1m} \\ \vdots & \ddots & \vdots \\ y_{\tilde{N}1} & \cdots & y_{\tilde{N}m} \end{bmatrix} \tag{4}$$

The objective function of ELM aims to simultaneously minimize the training error and the norm of the output weights, which can be mathematically represented as

$$\begin{aligned} &\min : \tfrac{1}{2}\|\boldsymbol{\beta}\|^2 + \tfrac{1}{2}C\sum_{i=1}^{\tilde{N}} \boldsymbol{\xi}_i^2 \\ &s.t., \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{y}_i - \boldsymbol{\xi}_i, i = 1, ..., \tilde{N} \end{aligned} \tag{5}$$

where $C$ denotes a regularization factor for generalization performance improvement, and $\boldsymbol{\xi}_i = [\xi_{1,m}, ..., \xi_{\tilde{N},m}]$ denotes the training error of the $m$ output nodes with respect to the training sample $\mathbf{x}_i$.

Then, based on the Karush-Kuhn-Tucker (KKT) theorem, we have

$$\tilde{\boldsymbol{\beta}} = \begin{cases} \mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{Y}, & N < L \\ \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{Y}, & N > L \end{cases} \tag{6}$$

where $\tilde{\boldsymbol{\beta}}$ is the estimated value of $\boldsymbol{\beta}$, and $\mathbf{I}$ is the unit matrix.

In addition, ELM kernel matrix is also defined [18]:

$$\Omega = \mathbf{H}^T\mathbf{H} : \Omega_{ij} = k\left(\mathbf{x}_i, \mathbf{x}_j\right) \tag{7}$$

where $k(\cdot, \cdot)$ is the inner product in the ELM feature space.

Therefore, kernel-based ELM can be represented as

$$f(\mathbf{x}) = \begin{bmatrix} k(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ k(\mathbf{x}, \mathbf{x}_{\tilde{N}}) \end{bmatrix}^T \left(\frac{\mathbf{I}}{C} + \boldsymbol{\Omega}\right)^{-1}\mathbf{Y} \tag{8}$$

**Remark 2.1:** Some conventional random projection belongs to linear mapping, but the feature mapping of ELM uses different kinds of nonlinear piecewise continuous neurons in the hidden layer. In addition, the hidden nodes of ELM are not only neurons with activation functions, such as Sigmoid function and Gaussian function, but also fuzzy rules, Fourier series, or wavelets, etc. [19, 20, 21].

**Remark 2.2:** One of the most important advantages of ELM is that its hidden layer parameters can be randomly generated according to any continuous probability distribution, and the universal approximation capability of "without iterative tuning" learning mode has been proved, which can guarantee the learning capability of ELM [22]. However, there is no corresponding theoretical proof for Schmidt's method, as well as lacks rigorous proof for full random hidden nodes cases of other random neural networks [23, 24, 25, 26, 27].

**Remark 2.3:** ELM can provide an unified learning paradigm for regression, binary classification, and multi-class classification, and be easily extended to both semi-supervised learning and unsupervised learning, as well as hierarchical structure (For example, support vector machine (SVM) is difficult to be used for dealing with multi-class classification directly. Usually, one-against-one or one-against-all strategy is implemented to convert multiclass classification into binary classification, but this kind of conversion may cause data imbalanced distribution, which degrades the performance of SVM classifier) [28, 29, 30, 31, 32]. In addition, all the hidden nodes in ELM are not only independent of training data, but also independent to each other. Thus, compared with other machine learning approaches, ELM is closer to biological learning.

## 3. Stacked ELM-AEs

The rapid development of both hardware and software accelerate the breakthroughs of deep learning, especially in image recognition, and natural language processing, etc. [33, 34, 35, 36, 37, 38, 39, 40]. The core reason is that deep learning approaches are deeper than shallow machine learning approaches, making neural network with hierarchical structure automatically extract high-level or representative features. It is well known that neural network can approximate any target function with enough number of hidden nodes, but too many hidden nodes also may lead to the overfitting problem, which seriously degrades its performance. Unfortunately, neural network with single hidden layer should be with a large number of hidden nodes to guarantee its performance especially in dealing with complex tasks. Hierarchical structure of deep neural network does not need a large number of hidden nodes in each layer, and its representation capability can be strengthened through layer-by-layer iterative strategy [41, 42]. However, because all the hidden layer parameters need to be fine-tuned multiple times, constructing deep neural network is too cumbersome, time-consuming, and may bring much human intervention, which always puzzles conventional deep learning. Accordingly, compared with existed shallow machine learning approaches, classic ELM and its variants can obtain comparative generalization performance and much faster learning speed. Thus, constructing deep neural network based on ELM theory or incorporating ELM into existed hierarchical structure of deep learning should be a promising way to tackle the above issues [43]. Among the ML-ELMs, stacked ELM-AE is most widely studied both in theory and applications.

In this section, we will review the stacked ELM-AEs, including general forms of stacked ELM-AE, denoising stacked ELM-AE, semi-supervised and unsupervised stacked ELM-AE, distributed stacked ELM-AE, and other typical variants of stacked ELM-AE.

### *3.1. General Forms of Stacked ELM-AE*

Autoencoder (AE) is an unsupervised neural network, whose history can be tracked back to 1980s [44, 45]. A classic AE is a back propagation (BP)-based neural network, in which the original input data are reconstructed at the output, passing through an encoding layer with less number of hidden nodes. AE aims to learn dense representation of the input data while maintaining most of the important information. Specifically, the input data are firstly mapped to an abstract feature space for representation, and then converted back into the original format. It can extract discriminative features and filter useless information from the input data by minimizing the reconstruction error.

As mentioned above, classic AE works based on BP algorithm, it is inevitable to inherit the drawbacks of gradient-based neural network, such as local optimum, time-consuming iterative process, and much human intervention, etc., which may degrade the performance of AE-based deep neural networks. Differently, ELM-AE was constructed based

on ELM theory, whose basic structure is illustrated in Fig. 2. For a given dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\tilde{N}} \subset R^n \times R^m$, the input data will be reconstructed at the output layer by

$$\sum_{i=1}^{L} \beta_i g(\mathbf{x}_i, \mathbf{a}_i, b_i) = \mathbf{x}_i \tag{9}$$

Similar to (2), the corresponding matrix form is

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{X} \tag{10}$$

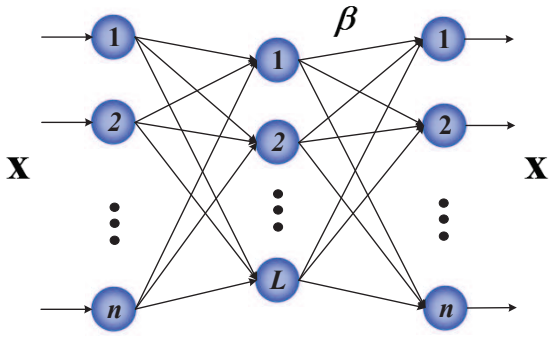



Figure 2: Basic Structure of ELM-AE

ELM-AE can represent features from the raw high-dimension space to the lower-dimension feature space, from the raw low-dimension space to the higher-dimension feature space, or the dimension of the raw space equals to the feature space. Accordingly, $\boldsymbol{\beta}$ can be achieved through the following three manners:

1) Compressed representation:

$$\boldsymbol{\beta} = \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{X} \tag{11}$$

2) Sparse representation:

$$\boldsymbol{\beta} = \mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{X} \tag{12}$$

3) Equal representation:

$$\boldsymbol{\beta} = \mathbf{H}^{-1}\mathbf{X} \tag{13}$$

**Remark 3.1:** Similar to multilayer perceptron (MLP), ELM-AE is a kind of SLFN. The essential difference between them is that ELM-AE aims to approximate the input data at the output layer, while MLP is for prediction with the given input data. In addition, both of them can be used for constructing neural networks with hierarchical structure. MLP with two hidden layer can deal with XOR problem, but its capability in feature extraction is not strong enough. Stacked ELM-AE can extract high-level feature well through its multiple hidden layers.

**Remark 3.2:** Some research works pointed out that local optimum of classic AE is not a serious issue for large hierarchical neural networks in dealing with practical applications. Furthermore, the landscape is packed with a great number of saddle points where the gradient is zero, and the surfaces curves up in most dimensions, this also can strengthen the performance of classic AE-based deep neural networks. We argue that this viewpoint is only based on the engineering experience, but not the rigorous theoretical support

Using ELM-AE to stack deep neural networks, i.e., stacked ELM-AEs, is one of most important branches in ELM-based deep learning. The main advantages of stacked ELM-AEs include: 1) ELM-AE inherits the very fast learning speed of classic ELM, which can reduce the time consumption of training process comparing with conventional deep learning approaches; 2) representation learning and decision making can be integrated into a whole training process,

in which multiple hidden layers stacked using ELM-AEs are for representation learning, and a final layer of ELM or an ELM classifier is implemented at the last component for decision making.

**Remark 3.3:** Different from conventional deep neural networks, stacked ELM-AEs do not need iterative fine-tuning after once all the parameters are fixed in each layer. Thus, its training time is reduced from days or hours to minutes or even seconds. The very fast learning speed of stacked ELM-AEs is especially useful in some real-world applications with strong timeliness, e.g., stock forecast and weather forecast, etc.

Fig. 3 illustrates the training of stacked ELM-AEs, ELM-AEs are performed for determining the output weights of the multiple hidden layers iteratively. Specifically, for the determination of the output weights between the $i$th hidden layer and the $(i+1)$th hidden layer, the number of input nodes of the $(i+1)$th ELM-AE should be identical to the hidden nodes of the $i$th hidden layer. The corresponding output of hidden layer in stacked ELM-AEs is

$$\mathbf{H}_{i+1} = g\left(\mathbf{H}_i \cdot \boldsymbol{\beta}_{i+1}^T\right) \tag{14}$$

where $\mathbf{H}_{i+1}$ and $\mathbf{H}_i$ represent the output matrices of the $(i+1)$th hidden layer and the $i$th hidden layer, and $g(\cdot)$ represents the activation function, respectively.

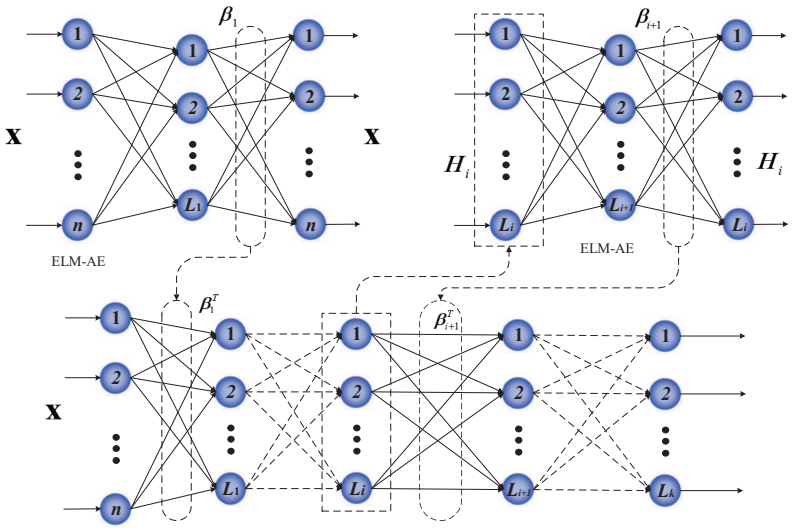



Figure 3: Typical Structure of Stacked ELM-AEs

Kasun et al. [46] proposed the first ELM-AE, which can represent features with singular values. In the proposed ELM-AE, the randomly generated hidden layer parameters, including input weights and bias, are required to be orthogonal, which can be mathematically represented as

$$\begin{aligned} &\sum_{i=1}^{L} \beta_i g(\mathbf{x}_i, a_i, b_i) = \mathbf{x}_i \\ &s.t., a^T a = \mathbf{I}, b^T b = 1 \end{aligned} \tag{15}$$

After that, Kasun et al. [46] designed an ELM-based deep neural network by stacking the proposed ELM-AEs, denoted by ML-ELM-1. Compared with DBN, DBM, stacked AE (SAE), and stacked denoising AE (SDAE), the proposed ML-ELM-1 achieved comparative accuracy with much faster learning speed. In addition, Cecotti [47] evaluated the performance of ML-ELM-1 on four handwritten character datasets, the corresponding experimental results also confirmed the advantages of ML-ELM-1 both in accuracy and time consumption. Actually, the hidden layer parameters in this ELM-AE are not purely randomly generated, because they are required to be orthogonal. Obviously, it has not well exploited the advantages of ELM, because Huang et al. [22, 28, 29] had pointed out that the universal approximation capability of ELM cannot be guaranteed without random projection of the inputs. Thus, we conjecture that the potential of ELM in the aforementioned hierarchical structure has not been fully exploited.

**Remark 3.4:** In this paper, ML-ELM denotes ELM-based deep neural networks, mainly including stacked ELM-AE, residual ELM, and local receptive filed based ELM (ELM-LRF), etc., and ML-ELM-1 denotes the multilayer ELM proposed by Kasun et al. in [46].

Recently, Wong et al. [48] proposed the kernel version of ML-ELM-1, named multilayer kernel ELM (ML-KELM), for strengthening the performance of ML-ELM-1. Different from ML-ELM-1, ML-KELM is stacked by kernel ELM-AE (KELM-AE), so the burden on tuning number of hidden nodes is eliminated by replacing the original random generated hidden layer parameters utilizing kernel matrix. Thus, ML-KELM should be more efficient and has better generalization performance. In addition, the transformation matrix can be learned through exact inverse rather than pseudoinverse, which can reduce the effects of reconstruction error on ML-KELM. Due to the kernel trick in ML-KELM, its training time mainly depends on the number of training data, while ML-ELM-1 depends on the number of hidden nodes. Accordingly, ML-KELM has obvious advantage in dealing with small-scale datasets in time consumption. Although, ML-KELM can tackle the issues of unstable and suboptimal performance of hidden layers of ML-ELM-1 caused by random projection and manual tuning of number of hidden nodes, it also needs a large memory and gradually becomes slow during the training process. Therefore, Vong et al. [49] proposed an extended version of ML-KELM, named ML-EKM-ELM, in which an approximate empirical kernel map (EKM) computed from low-rank approximation of the kernel matrix was used for producing much smaller hidden layers for fast training and low memory storage. Comprehensive experiments indicate the effectiveness of the proposed ML-EKM-ELM, and it is more suitable for large-scale problems compared with ML-ELM-1 and ML-KELM.

In order to tackle the above issues and further strengthen the performance of ML-ELM-1, Tang et al. [50] designed a $\ell_1$-norm ELM-AE and attempted to use it as the building block of a hierarchical ELM (H-ELM), in which the hidden layer parameters of $\ell_1$-norm ELM-AE do not require to be orthogonal. The modified objective function can be represented as

$$\min : \|\boldsymbol{\beta}\|_{\ell_1} + \|\mathbf{H}\boldsymbol{\beta} - \mathbf{X}\|^2 \tag{16}$$

Due to the utilization of $\ell_1$ penalty, the new ELM-AE could achieve sparser and more meaningful features, and the proposed H-ELM achieved better accuracy and much faster convergence speed than some conventional deep learning approaches as well as ML-ELM-1 on car detection, gesture recognition, and online incremental tracking. Different from ML-ELM-1, a classic ELM was applied for the final decision making in H-ELM.

**Remark 3.5:** $\ell_1$ optimization has been proved that it can function better in data recovery and other applications [51, 52], so the $\ell_1$-norm ELM-AEs can reduce the redundant features and remove noise, as well as accelerate the convergence of H-ELM. However, there is no analytical solution for this kind of sparse ELM-AE and solving the optimal output weights have to resort to the $\ell_1$-norm based optimization algorithms, such as the fast iterative shrinkage-thresholding algorithm used in [50].

Considering ML-ELM may be sensitive to noise and outliers, Chen et al. [53] proposed a full correntropy-based ML-ELM (FC-MELM), in which both of the loss function and the sparsity penalty term in conventional ELM-AE were replaced by correntropy-based loss function and correntropy-based sparsity penalty, respectively. Thus, the proposed FC-MELM stacked by correntropy-based ELM-AEs is more robust and can provide sparser representation compared with H-ELM.

In order to solve large and complex tasks using ELMs without incurring the memory problem, Zhou et al. [54] proposed stacked ELM (S-ELM) with hierarchical structure based on principal component analysis (PCA). As the hidden layer parameters of ELM-AE are randomly generated, the contribution of different hidden nodes to performance improvement may vary a lot. It means that there are some redundant hidden nodes in ELM-AE, which not only degrade the performance, but also increase the computational burden. S-ELM selects the most significant few percent of hidden nodes or combined hidden nodes to represent all the hidden nodes in each ELM-AE. Specifically, in the hierarchical structure, the previous layer outputs such hidden nodes to the next layer, and those hidden nodes are then combined with the new randomly generated hidden nodes and function as the total hidden layer output of this layer. During its training process, the above procedure as the previous layer can output the most significant hidden nodes to the next layer, making S-ELM always keep a compact network size. In addition, S-ELM selects top few significant hidden nodes using their output weights' eigenvalues, and reduces the number of hidden nodes by multiplying the corresponding eigenvectors based on PCA.

**Remark 3.6:** Similar to ML-ELM-1 and H-ELM, S-ELM consists of multiple small ELMs located at different

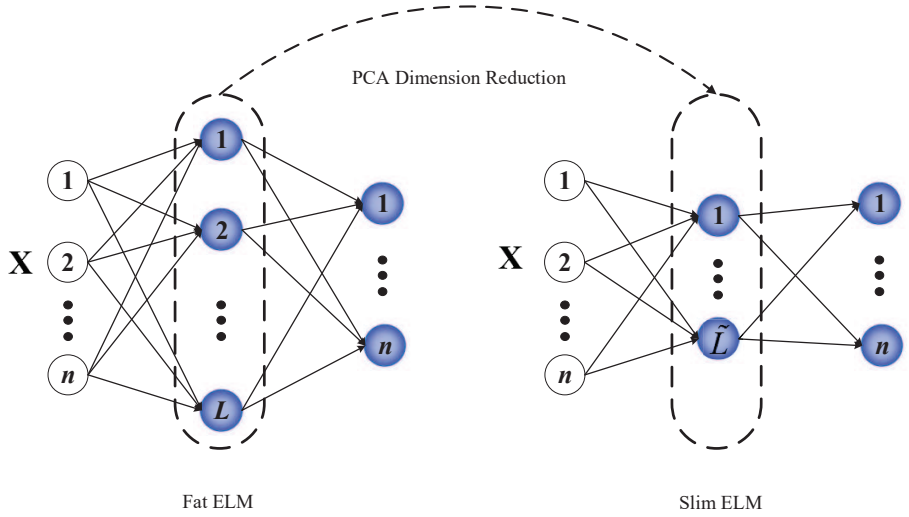



Figure 4: Fat ELM is reduced to a slim ELM in S-ELM

layers, each of them is serially connected. Differently, it is not the original full hidden layer outputs that are propagated to the next layer, but the hidden layer reduced by PCA.

As shown in Fig. 4, the original "fat" ELM with $L$ hidden nodes is reduced to a new "slim" ELM with $\tilde{L}$ hidden nodes through PCA. The reduced hidden layer output matrix $\tilde{\mathbf{H}}_1$ is propagated to the next layer to represent the information of all the original hidden nodes of this layer. Accordingly, the second layer will randomly generate $\left(L-\tilde{L}\right)$ hidden nodes, and the corresponding hidden layer output matrix is denoted as $\mathbf{H}_{2new}$. The actual hidden layer output matrix of the second layer is

$$\mathbf{H}_2 = \left[\tilde{\mathbf{H}}_1, \mathbf{H}_{2new}\right] \tag{17}$$

In the following layers of S-ELM, the same PCA dimension reduction procedure will be taken iteratively until to the last layer. The whole dimension reduction and feature mapping process can be represented as

$$\begin{gathered}
\mathbf{H}_1 \rightarrow \tilde{\mathbf{H}}_1 \\
\left[\tilde{\mathbf{H}}_1, \mathbf{H}_{2new}\right] \rightarrow \tilde{\mathbf{H}}_2 \\
\left[\tilde{\mathbf{H}}_2, \mathbf{H}_{3new}\right] \rightarrow \tilde{\mathbf{H}}_3 \\
\vdots \\
\left[\tilde{\mathbf{H}}_{(N-2)}, \mathbf{H}_{(N-1)new}\right] \rightarrow \tilde{\mathbf{H}}_{(N-1)} \\
\left[\tilde{\mathbf{H}}_{(N-1)}, \mathbf{H}_{Nnew}\right]
\end{gathered} \tag{18}$$

Comprehensive experiments indicate that S-ELM can achieve better or comparative accuracy but with compact network size than ELM, SVM, and DBN.

Although S-ELM can address large and complex data problems with a relatively high accuracy and low requirement for memory, there is still room for improving the time consumption and robustness. Luo et al. [55] enhanced S-ELM by replacing the original PCA with the correntropy-optimized temporal PCA (CTPCA), which is more robust for outlier rejection and can reduce the training time significantly.

Considering the limited capability of shallow one-class ELM (OC-ELM) in dealing with high-dimension and complex tasks, Dai et al. [56] proposed multilayer neural network based one-class classification with ELM (ML-OCELM) for learning acceleration and performance enhancement. In ML-OCELM, ELM-AEs are for feature extraction and OC-ELM is implemented at the final phase for one-class classification. With the encoded features, OC-ELM can

perform the final decision making based on the following objective function:

$$\min : \|\mathbf{H}_{\mathbf{X}^{(k)}}\boldsymbol{\beta} - \mathbf{Y}\|_2^2 + \|\boldsymbol{\beta}\|_2^2 \tag{19}$$

where $\mathbf{H}_{\mathbf{X}^{(k)}}$ denotes the hidden layer output matrix with the encoded feature $\mathbf{X}^{(k)}$ as the input feature.

In MK-OCELM, setting of number of hidden nodes is not necessary due to the kernel trick in ELM-AE. Thus, MK-OCELM is with less human-intervention parameters tuning and better generalization performance.

In [57], we stacked ELM-AEs for designing a multilayer probability ELM (MP-ELM). Different from the aforementioned ML-ELMs, MP-ELM outputs the probability of the predicted results belonging to all the classes instead of fitting to data, which can significantly alleviate the effects of accumulated errors on the final predicted results.

### *3.2. Denoising Stacked ELM-AE*

Motivated by [58, 59], Zhang et al. [60] proposed ELM denoising AE (ELM-DAE) by introducing a local denoising criterion. Different from ELM-AE, the inputs and the outputs of ELM-DAE are initial training data and corrupted data, respectively. The authors used Gaussian noise, Masking noise or Salt-and-pepper noise to corrupt the initial training data. In this way, features extracted by ELM-DAE should be more robust. Sequentially, denoising ML-ELM (D-ML-ELM) was stacked by ELM-DAE. Furthermore, manifold regularization framework was introduced into D-ML-ELM and denoising Laplacian ML-ELM (D-Lap-ML-ELM) was proposed. Due to the utilization of manifold regularization term in the objective function of D-Lap-ML-ELM, it can introduce the local manifold structure information of the data, which is regarded as a prior knowledge into the classification model for enhancing the performance. The objective function of D-Lap-ML-ELM can be mathematically represented as

$$\min : \|\boldsymbol{\beta}\|^2 + \|\mathbf{H}\boldsymbol{\beta} - \mathbf{X}\|^2 + \mathrm{Tr}\left(\boldsymbol{\beta}^T\mathbf{H}^T\mathbf{L}\mathbf{H}\boldsymbol{\beta}\right) \tag{20}$$

where $\mathrm{Tr}(\cdot)$ stands for the trace of a matrix, and $\mathbf{L}$ is the Laplacian matrix.

Experimental results indicate the effectiveness of D-ML-ELM and D-Lap-ML-ELM both in supervised learning and unsupervised learning on some typical benchmark datasets.

**Remark 3.7:** D-Lap-ML-ELM needs to calculate the Laplacian matrix, so its computational burden is higher than D-ML-ELM, especially when the number of unlabeled data is relatively large.

Similar to ELM-DAE, Cao et al. [61] proposed a SSDAE-RR (stacked sparse denoising AE-ridge regression) learning scheme, integrating sparse denoising stacked AE and ridge regression implementation in ELM. The proposed SSDAE-RR uses a quick-and-dirty SSDAE to generate a stable and interpretable feature space which is fed into a RR solver in ELM to calculate the output weights. Experimentally, the time consumption of SSDAE-RR is comparative with ELM, as long as its embodied deep neural network only needs a few iterations for unsupervised pre-training.

### *3.3. Semi-Supervised and Unsupervised Stacked ELM-AE*

Hu et al. [62] stacked a deep neural network based on unsupervised ELM [63] (named as unsupervised SLFNs with randomly fixed hidden neurons, URHN-SLFNs, and the proposed stacked deep neural network was named as St-URHN-SLFNs), whose structure could yield a better embedding space for clustering problem. Similar to (20), the objective function of the proposed URHN-SLFNs is

$$\min : \|\boldsymbol{\beta}\|^2 + \mathrm{Tr}\left(\boldsymbol{\beta}^T\mathbf{H}^T\mathbf{L}\mathbf{H}\boldsymbol{\beta}\right) \tag{21}$$

The proposed St-URHN-SLFNs not only incorporates the simplicity of URHN-SLFNs, but also inherits the excellent representation capability derived from hierarchical structure. Compared with Laplacian eigenmaps, spectral clustering, and K-means clustering, its training time is relatively longer, but it only needs several seconds at most. Compared with deep AE and SAE, its learning speed is still much faster.

Sun et al. [64] also proposed an unsupervised ELM-AE by combining manifold regularization with ELM-AE, named generalized ELM-AE (GELM-AE), which outperformed some state-of-the-art unsupervised learning approaches, including Laplacian embedding, spectral clustering, and K-means clustering, etc. In addition, they stacked a deep neural network using GELM-AEs, named as multilayer generalized ELM-AE (ML-GELM).

Gu et al. [65] proposed a semi-supervised deep ELM (SD-ELM) for WiFi indoor localization. In the proposed SD-ELM, discriminative features are fed into a classifier after feature extraction, and its objective function is similar to (20) and (21).

### 3.4. Distributed Stacked ELM-AE for Big Data

According to analyzing the execution process of H-ELM, Chen et al. [66] found that H-ELM was still relatively time-consuming in some sub-processes. Thus, they proposed a parallel H-ELM (PH-ELM) on Flink to accelerate H-ELM based on three basic parallel algorithms. As shown in Fig. 5, the executions of both sparse ELM-AE layers and the classic ELM are parallelized on Flink. Before the training process, the data firstly need to be stored in HDFS, and then, it is loaded into Flink's distributed memory system as a DST object. Simultaneously, the hidden layer parameters of all the ELM-AEs and the classic ELM are randomly generated. Finally, all the output weights of all the ELM-AEs and the classic ELM are calculated and stored in Flink's distributed memory system as DST objects, and they also can be written as HDFS.

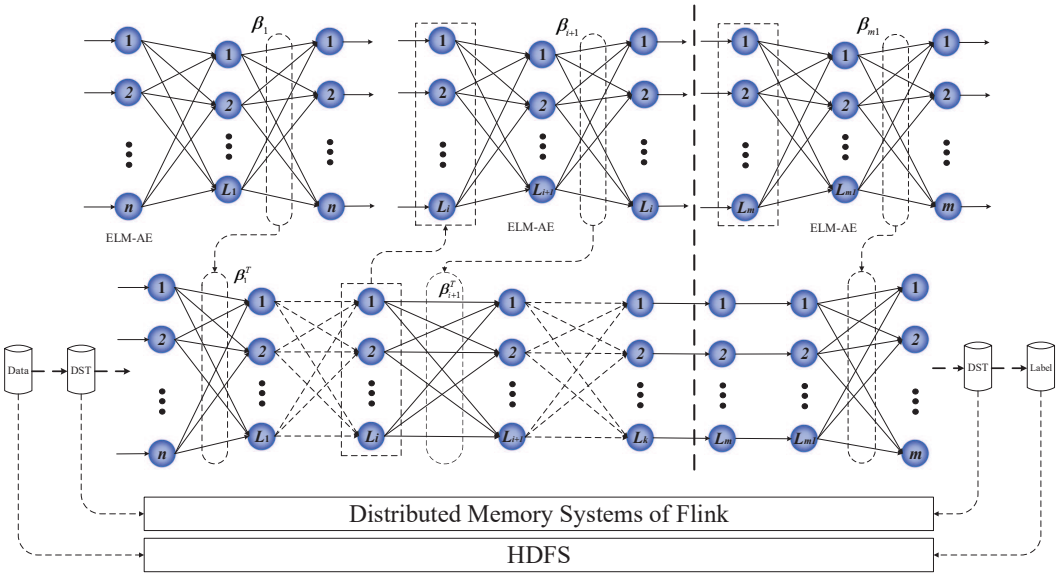



Figure 5: Workflow of PH-ELM

Yao et al. [67] analyzed the implementation of ELM and H-ELM on MapReduce as the distributed and parallel models for process quality prediction with big data. Accordingly, they proposed distributed and parallel H-ELM (dp-HELM), in which hidden layers of H-ELM were decomposed into a loop of MapReduce jobs. Under the circumstance of large-scale dataset, the multimode process was considered and the dp-K-means algorithm was proposed for dividing the process into a group of modes. Then, the local models of different modes were trained concurrently by dp-ELM and dp-HELM models, respectively. Finally, the local modes were integrated for online prediction. Experimentally, the proposed MapReduce-based dp-HELM has shown excellent capability in processing and extracting information from big data.

### 3.5. Other Variants of Stacked ELM-AE

Tissera et al. [68] proposed a supervised AE ELM module, which could be utilized for stacking deep neural network. Fig. 6 illustrates the first two hidden-layer of the proposed deep ELM network. In this process, the input data are firstly projected to the $L$-dimension first hidden-layer through a random weight matrix. After transformation by the sigmoidal units, the result is multiplied by the output weights to produce a new input vector. Sequentially, the input vector is projected to a $Q$-dimension second hidden-layer through another random weight matrix. The authors referred to each three-layer ELM as a module, and additional ELM modules could be added, with a readout of a classification available from each intermediate layer, to construct a deep neural network.

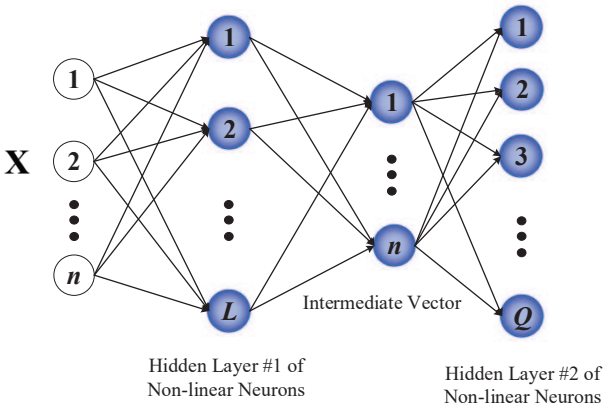



Figure 6: First Two Hidden-Layers of the Supervised Deep ELM Network

Multimodal data are becoming ubiquitous in the daily life, but it is difficult to ensure the data collected from different sources are full pairing. In order to tackle this issue, Wen et al. [69] proposed a modified framework of weakly paired multimodal fusion based on ML-ELM-1, which could find complex nonlinear transformations of each modality of data such that the resulting representations were highly corrected. In the proposed framework, ML-ELM-1 implements for extracting features of all the modalities separately, and the extracted discriminative features are performed joint dimension reduction by weakly paired maximum covariance analysis. Compared with linear weakly paired approaches, the proposed framework can achieve better performance with better robustness.

Chu et al. [70] proposed a network embedding-based deep ELM, (DELM-NE), which was stacked by DELM-AEs. As shown in Fig. 7, DELM-AE consists of several hidden layers, and the parameters between the last ELM-based hidden layer and the encoder output can be obtained by fine-tuning using BP algorithm. According to the structure illustrated in Fig. 7, compared with BP-based AE, most of the parameters in DELM-AE need not to be fine-tuned, leading to high-increased efficiency; and compared with conventional ELM-AE, the fine-tuning in the last part of DELM-AE can reduce the negative effects of random projection, so it should be with better generalization performance. Experimental results on some benchmark datasets indicate the excellent performance of the stacked deep ELM using DELM-AE.

Different from existed ELM-AEs, in which the hidden nodes in the encoding layer are randomly generated, possibly leading to the suboptimal feature mapping, Yang et al. [71] established a two-layer ELM-AE, whose current weights of the encoding layer were replaced by the previous decoding layer and were very correlated with the input data, making it naturally symmetric. In addition, they stacked a ML-ELM using the proposed two-layer ELM-AE, and implemented for dimension reduction and image reconstruction.

Yang et al. [72] also proposed a general structure of ML-ELM with subnetwork nodes, providing a representation learning platform with unsupervised/supervised and compressed/sparse representation learning. They found that a hidden node itself could be a subnetwork formed by several hidden nodes which naturally formed biological learning, and thus resulted in feature learning (See Fig. 8). It means that a single hidden layer can contain multiple networks. Different from classic ELM, subnetwork nodes (also called general hidden nodes) are performed instead of single hidden nodes. Similar to classic ELM, the number of general hidden nodes and output dimension are also independent, but the number of hidden nodes in each general hidden node should be equal to the dimension of outputs, i.e., the number of hidden nodes in a general hidden node should equal the number of output nodes.

**Remark 3.8:** As mentioned above, all the hidden layers of other existed ELM-AEs randomly generate their hidden layer parameters, but it cannot function well in dealing with some specific tasks sometimes, because the pure random projection may destroy the useful features. Some research works have pointed out that ELMs are sensitive to the random range of the hidden layer parameters [73, 74].

**Remark 3.9:** According to the experimental results in [72], ML-ELM stacked by subnetwork ELM is not sensitive to generalization performance. The proposed approach provides a new forward encoding learning way that is different from other existed stacked ELM-AEs and conventional deep neural networks, which may be a new direction of

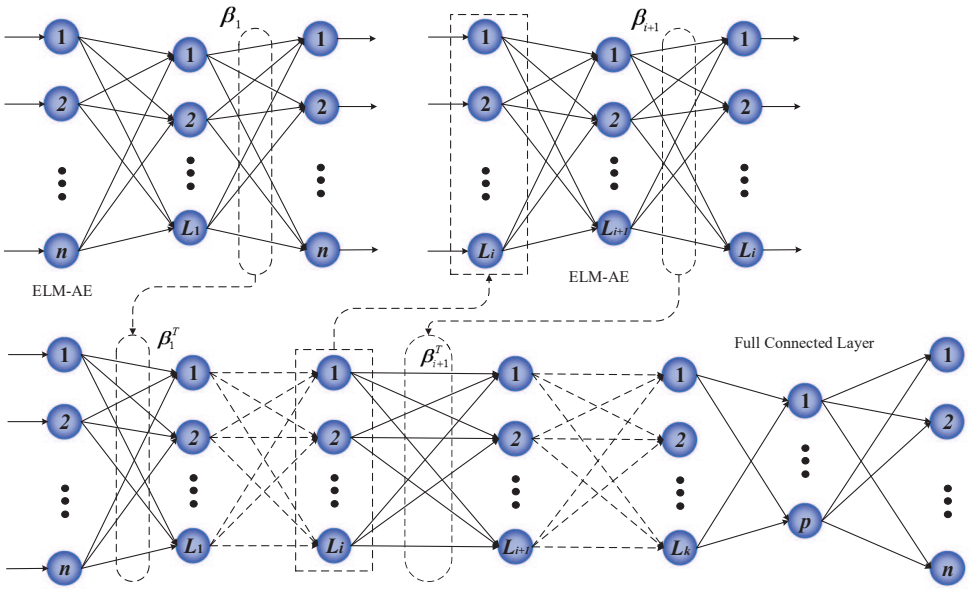
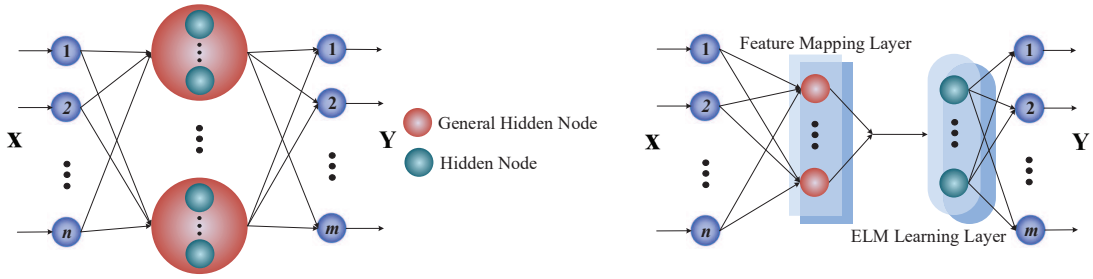



Figure 7: Structure of DELM-AE

representation learning. Other related research works about the subnetwork node refer to [75, 76, 77, 78, 79].




(a) Feature Mapping of ELM with Subnetwork Nodes

(b) Subnetwork ELM for Representation Learning

Figure 8: Basic Structure of Subnetwork ELM

We noticed some research works about the online sequential ELM-AE (OS-ELM-AE) based on the online version of ELM [80, 81], i.e., online sequential ELM (OS-ELM) [82, 83, 84]. Mriza et al. [80] and Su et al. [81] verified the performance of multilayer OS-ELM (ML-OSELM) in image classification and hot metal silicon content prediction, respectively. Similar to OS-ELM, the proposed ML-OSELM can learn the sequentially coming data one-by-one or chunk-by-chunk with fixed or varied chunk size. As we all known, deep neural network usually needs a great number of training data to guarantee its performance. If the specific chunk is with small number of data, and without sequentially coming chunks, it may be difficult to guarantee the performance of ML-OSELM. In addition, we conjecture that ML-OSELM is difficult to guarantee its stability with the data stream, because it needs to update its relatively complex structure in every time step with random projection. Accordingly, we envision OS-ELM-AE will be one of the hot research topics, which still has many challenging issues should be tackled before ML-OSELM can be really applied.

## 4. Residual ELMs vs. Deep Residual Networks

Deep residual network (ResNet) was proposed for easing the training of deep neural networks, in which stacked layers were fitted residual mapping instead of desired underlying mapping [85, 86, 87]. However, Veit et al. [88] found that short paths of ResNet behaved just like the ensembles without strongly depending on each other. Differently, the residual compensation ELM (RC-ELM) was proposed based on hierarchical residual compensation mechanism, in which the baseline ELM was for building the feature mapping between the input and the output, and the following ELMs were for residual compensation layer by layer through remodeling the residual of the previous layers for performance improvement, as illustrated in Fig. 9 [89]. In addition, the coupling relationship among ELM modules has been proofed, indicating that RC-ELM is not a stack of several nonlinear modules. Compared with classic ELM, RC-ELM has better performance in dealing with regression problems, because the ELM modules in RC-ELM can enhance its representation capability iteratively. For example, it may lead to overfitting problem when classic ELM uses a large number of hidden nodes to enhance its representation capability, but that number of hidden nodes can be shared by ELM modules in RC-ELM, which can perform the deep compensation of residual layer by layer to obtain the optimal solution, so it can avoid the overfitting problem efficiently without the loss of representation capability.

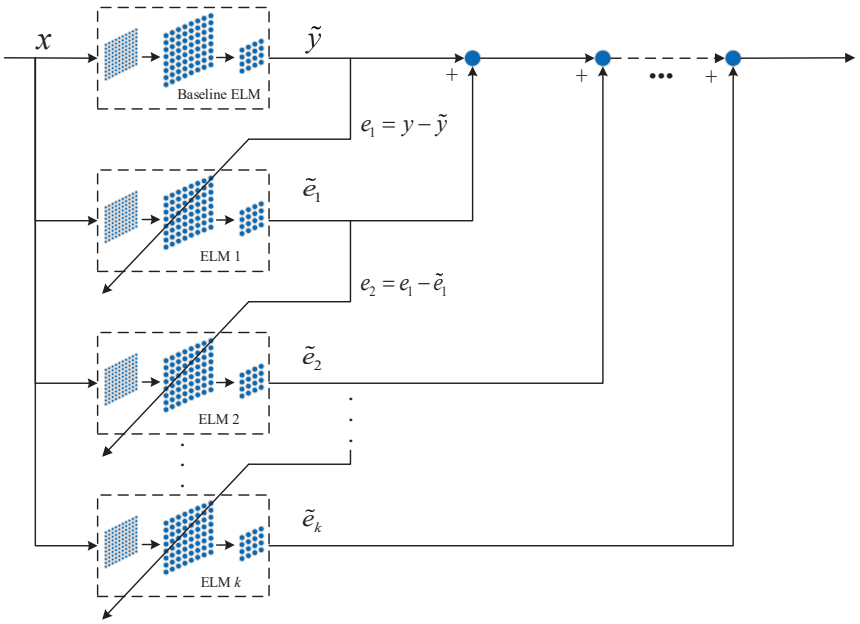



Figure 9: Structure of RC-ELM

In order to further strengthen the capability of RC-ELM in dealing with non-Gaussian noise, we proposed an extended version of RC-ELM [90]. Compared with RC-ELM, the extended version abandons the expansion in depth, and performs expansion in broad sense, because the experimental results in [89] indicate that the first residual compensation layer usually plays a main role in performance improvement. It was designed as a two-layer structure, including one baseline layer and one residual compensation layer, but the residual would be decomposed into some sub-modes using empirical wavelet transform (EWT), and corresponding number of ELMs were implemented for modeling those sub-modes in the residual compensation layer (see Fig. 10). The extended version is more efficient in handling tasks with non-Gaussian nosie, because the decomposed sub-modes are more linear.

Similarly, Tissera et al. [91] designed a hierarchal ELM based on a modular architecture, named modular expansion ELM (M-ELM), which expanded the output weight layer constructively, so that the final network could be treated as a SLFN with a "large" hidden layer. The training speed of M-ELM is much faster than a signal hidden layer neural network with less memory and computational complexity. Actually, RC-ELM and M-ELM have similar network structure, but RC-ELM is for regression problem, and M-ELM mainly for classification problem.

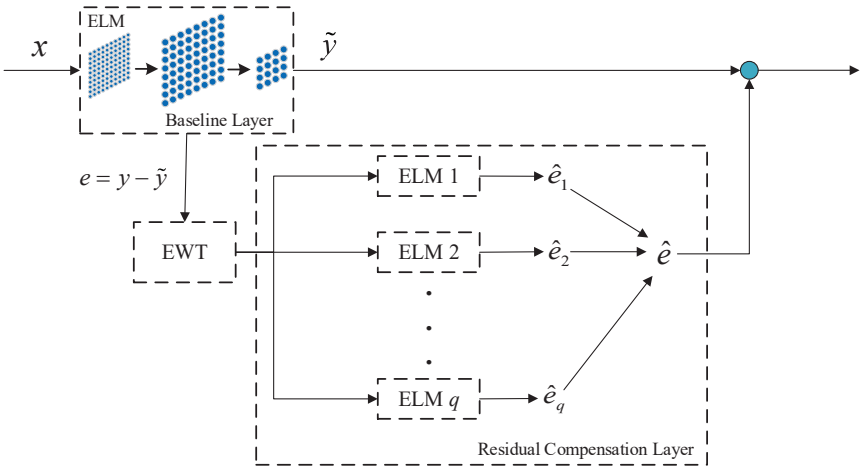



Figure 10: Structure of Extended Version of RC-ELM

Yu et al. [92] proposed a stacked structure, named deep representation learning via ELM (DrELM), in which a scheme of building layer-by-layer structure for estimating the errors of prediction functions when working on a particular learning set, and then correcting those errors. In each layer, DrELM integrates a random projection of the predictions obtained by classic ELM into the inputs, and then applies kernel functions to generate the outputs for the next layer. In this way, data from different classes are pushed towards different directions so that the resulting features are more likely to be separated.

In [93], Wang et al. firstly proposed an enhanced DrELM (EH-DrELM), in which the linear ELM was replaced by regularized ELM (R-ELM) as the building blocks, and adding shortcut connections between the inputs of two successive building blocks (similar to ResNet). After that, a modified AdaBoost-ID algorithm was proposed for updating the weights of both the correctly classified and misclassified samples. Finally, they embedded the modified AdaBoost-ID into EH-DrELM for developing a deep weighted ELM (DWELM) to deal with complex and large imbalanced data.

Yang et al. [94] proposed a parent-offspring progressive learning method (PPLM) for strengthening the representation capability and generalization performance of classic ELM, in which a partition growth method was firstly proposed to separate similar feature data into the same partition, and several ELMs were utilized to learn each corresponding partition. The proposed PPLM extends classic ELM from a single neural network to a multi-network learning system, and theoretical proof and experimental results indicate that it can approximate any target continuous function and classify disjointed regions with excellent performance. Similar to the above residual ELMs, the large number of hidden nodes in classic ELM can be shared by the multiple ELMs of PPLM, so it can reduce time consumption and avoid overfitting problem.

## 5. Local Receptive Field based ELM

Usually, the hidden nodes of ELMs are fully connected to the input nodes, which can produce excellent generalization performance. However, some applications, such as image processing and speech recognition, may include strong local correlations, and it is reasonably expected that the corresponding neural networks have local connections instead of full connections. Inspired by CNN, one of such local receptive fields may be implemented by randomly generating convolutional hidden nodes and the universal approximation capability of such ELM may still be preserved. Accordingly, Huang et al. [95] proposed the local receptive field based ELM (ELM-LRF) as a generic ELM architecture to tackle image processing and other related tasks in which different density of connections may be requested. The connections between the inputs and the hidden nodes are sparse and bounded by corresponding receptive fields, which can be sampled by any continuous probability distribution. In addition, combinatorial nodes are used for providing translational invariance to the network by combining several hidden nodes together. Compared with CNN, ELM-LRF does not involve gradient-based steps, so its training should be remarkably efficient. Bai et al. [96] used ELM-LRF as

a general framework for object recognition, which was operated directly on the raw images, and thus suitable for all different datasets.

Fig. 11 illustrates the local dense connection of ELM-LRF, in which the connections between the input layer and the specific hidden node are randomly generated depending on some continuous probability distribution, and some receptive fields are then generated through this kind of random connections.

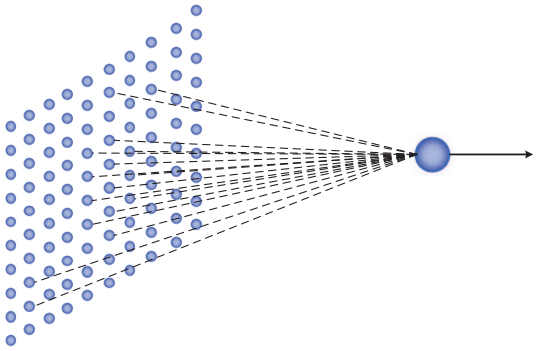


Figure 11: Local Connections of ELM Hidden Node

**Remark 5.1:** Combinatorial node means that the hidden node of ELM can be a combination of several hidden nodes or a subnetwork of nodes. Combinatorial node based ELM-LRF may learn the local structure better: denser connections around the input node due to the overlap of specific number of receptive fields while sparser farther away.

In detail, the receptive field of each hidden node consists of input nodes within a predetermined distance to the center. In addition, simply sharing the input weights to different hidden nodes directly leads to the convolution operation. In this manner, a specific case of the general ELM-LRF can be created, which is shown in Fig. 12.

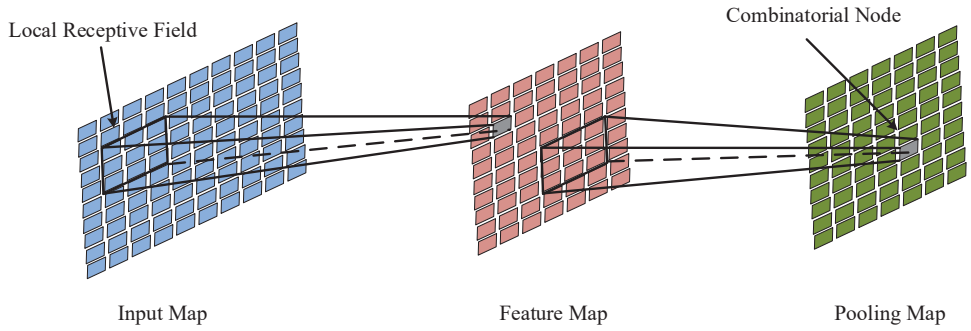



Figure 12: General Form of ELM-LRF

ELM-LRF is actually closely related to CNN, both of them handle the raw input directly and apply local connections to force the network to learn spatial correlations in natural images and languages. Differently, ELM-LRF can provide more flexible and wider type of local receptive fields, but CNN only uses convolutional hidden nodes; ELM-LRF randomly generates the input weights and analytically calculates the output weights, but CNN needs to be tuned.

Liu et al. [97] proposed a multi-modal ELM-LRF (MM-ELM-LRF) framework for constructing the nonlinear representation from different aspects of information sources, which has three separated phases, including unsupervised feature representation for each modality separately, feature fusion representation, and supervised feature classification. The authors performed feature learning to have representations of each modality, i.e., RGB and Depth, before they were mixed. Each modality was given to a single ELM-LRF, which could provide useful translational invariance of

low-level features, such as edges, and allowed parts of an object to be deformable to some extent. In this structure, MM-ELM-LRF takes full advantages of ELM-LRF to learn the high-level representation of the multi-modal data.

In [98], a hierarchical local-receptive-field-based ELM structure was proposed to jointly learn the state representation and the reinforcement learning strategy. As shown in Fig. 13, ELM-LRF was extended to a multilayer structure. In a single ELM-LRF, the links between the input and the hidden layer nodes are sparse and bounded by corresponding receptive fields, which are sampled from any continuous probability distribution. The feature maps in convolution layers are determined by the filters sliding on the previous layer pixel by pixel. The pooling layer is in full connection with the output layer, and the output weights are analytically calculated through least-square estimation. In addition, Li et al. [99] proposed a hierarchical ELM with LRF, whose structure is similar to the one illustrated in Fig. 13.

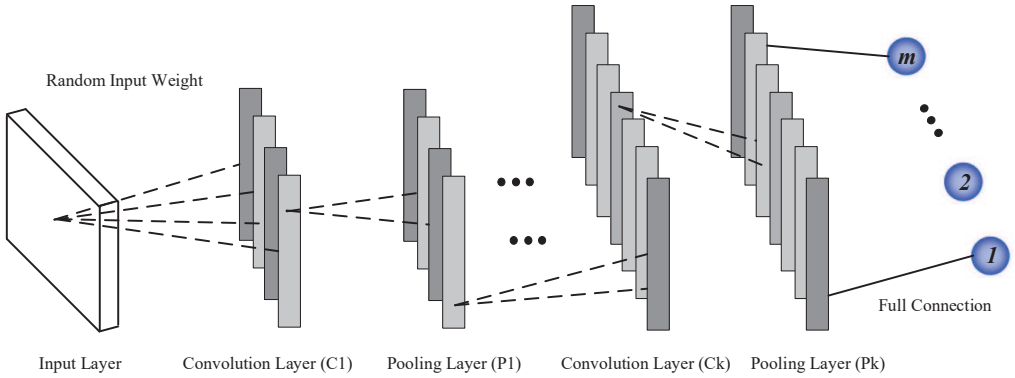



Figure 13: Structure of Hierarchical ELM-LRF

A modified ML-ELM was proposed for image classification with two stages, including ML-ELM feature mapping stage and ELM learning stage [100]. The ML-ELM feature mapping stage was recursively built by alternating between feature map construction and maximum pooling operation. In the ELM learning stage, elastic-net regularization was proposed to learn the output weights, which could guarantee to learn more compact and meaningful output weights.

Zhu et al. [101] designed a modified hierarchical ELM for unsupervised representation, which learned the local receptive filters by ELM-AE. In addition, several key elements were combined together to boost the performance, including local contrast normalization, whitening and trans-layer representation.

## 6. Applications

Due to the excellent training speed, generalization performance, and representation capability, ML-ELMs have been performed in many real world applications.

Wang et al. [102] performed ML-ELM-1 for encrypted image classification. Similarly, Ahmad et al. [103] exploited ML-ELM-1 for hyperspectral image classification. Cao et al. [104] also focused on hyperspectral image classification using modified ML-ELM, named multilayer sparse ELM (MSELM). Considering that the neighboring pixels are more likely from the same class, a local block extension was introduced for MSELM to extract the local spatial information, named local block MSELM (LBMSELM). Furthermore, the loopy belief propagation was also implemented in MSELM and LBMSELM for further using the rich spectral and spatial information to improve the classification accuracy. Considering the effectiveness of leaky rectified unit (LReLU) activation function, Nayak et al. [105] replaced the original activation function of ELM-AE using LReLU activation (ML-ELM+LReLU), and applied the proposed ML-ELM+LReLU for pathological brain image classification.

Yang et al. [106] proposed a modified hierarchical ELM-based image denoising network, which comprised a sparse AE and a supervised regression, also including a non-local aggregation procedure aiming to fine-tune noise reduction according to structural similarity. Yu et al. [107] proposed a ML-ELM with object principal trajectory, in

which the temporal and spatial characteristics were taken into consideration for supporting dynamic semantic representation between adjacent frames. The proposed approach can recognize multiple objects with different movement directions, and also identify subtle semantic features.

Duan et al. [108] designed a system for motor imagery electroencephalogram (EEG) classification, in which PCA and linear discriminant analysis (LDA) were combined for feature extraction, and ML-ELM-1 was performed for classifying. Experimentally, the designed ML-ELM-1-based system was more suitable in dealing with motor imagery EEG data. She et al. [109] proposed the hierarchical semi-supervised ELM (HSS-ELM) for motor imagery EEG classification, in which H-ELM was used for feature extraction, and semi-supervised ELM for the final classification. Furthermore, Kadam et al. [110] combined wavelet packet transform and H-ELM for EEG based IQ test. Yin et al. [111] proposed dynamic deep ELM (DD-ELM) to adapt the variation of the EEG feature distributions across two mental tasks for task-generic mental fatigue recognition. Compared with existed ML-ELMs, DD-ELM iteratively updated the shallow weights at multiple time steps during the testing stage, incorporating both the merits from deep neural network for EEG feature abstraction and ELM-AE for fast weight recomputation. Ding et al. [112] combined ML-ELM-1 and kernel ELM (K-ELM) for EEG classifcation, i.e., ML-ELM-1 for feature extraction and K-ELM for classification.

Niu et al. [113] used CNN and ML-ELM-1 as feature extractor, and K-ELM as the classifier for human activity recognition. Chen et al. [114] incorporated the kernel risk-sensitive loss into S-ELM for achieving fine-grained human activity recognition.

Ibrahim et al. [115] designed a three-stage framework, which combined PCA, deep ELM (DELM), and LDA, for protein fold recognition from the amino-acid sequences. They also designed another two frameworks for protein fold recognition [116]. In the first two-level framework, deep kernelized ELM (DKELM) and LDA were performed. In the second three-level framework, OVADKELM and OVODKELM were independently employed to extract features, and DKELM was used for the final classification.

Roul et al. [117] implemented ML-ELM-1 for text data classification, and obtained satisfactory stability and effectiveness. Cao et al. [118] proposed a modified approach for radar emitter signal identification, where the bispectrum estimation of radar signal was extracted and H-ELM was performed for further representation learning and recognition. Experimentally, four representative radar signals were conducted for performance validation, and the results indicated that the proposed approach was more feasible and potentially applicable in real world applications. Zhang et al. [119] proposed a modified ML-ELM classification model combined with dynamic generative adversarial net (GAN) for imbalanced biomedical data classification, in which PCA was used for removing irrelevant and redundant features, and GAN was designed to generate the realistic-looking minority class samples for balancing the class distribution, a self-adaptive ML-ELM was finally proposed for classification.

**Remark 6.1:** According to the aforementioned introduction and analysis, stacked ELM-AEs play very important roles in ELM-based deep learning filed, and they can obtain excellent performance in dealing with classification and recognition tasks, because the discriminative features can be extracted from the raw data through multiple hidden layers. However, if stacked ELM-AEs are implemented to deal with some regression problems, it only can obtain comparative results compared with shallow machine learning approaches, such classic ELM and SVM, not much better than them. The main reason may be that the extracted features by the multiple hidden layers are useless to the final predictions. For example, when we meet a stranger, we may deduce his gender, age, or job by observing him, because his appearance (features) should be helpful (this kind of situation is actually equal to the classification or recognition problems in machine learning and deep learning); however, we may not be able to accurately deduce where does he live or what will do only from his appearance (features) (this kind of situation is actually equal to the regression problems in machine learning and deep learning).

## 7. Discussions

ELM was originally inspired by biological learning and proposed to tackle the challenging issues faced by some existed machine learning approaches. Some related research works indicate that brain learning is usually sophisticated [120, 121, 122], and brain can function well for regression, classification, clustering, and feature extraction almost without human intervention and zero time in learning given particular samples in several scenarios. Accordingly, Huang et al. [28] conjectured that some parts of brain might be with random neurons with all their parameters

independent of environments, and the resultant learning mechanism was referred to ELM theory. Actually, the learning mechanism of ELM is globally ordered but locally random based, such kind of way can guarantee the universal approximation capability and classification capability. In this section, we will discuss about ELM combining with conventional deep learning, whether other random neural networks can be used for designing AE, and the randomization in conventional deep learning, respectively.

### *7.1. ELM Combining with Conventional Deep Learning*

In the common practices for the combination of ELM and conventional deep learning approaches, the conventional deep learning schemes are used for feature extraction, and ELM for the final decision making based on the extracted high-level features. For example, Ribeiro et al. [123] utilized DBN for feature extraction and ELM for classification. Han et al. [124] first produced an emotion state probability distribution for each speech segment using deep neural networks, then constructed utterance-level features from segment-level probability distributions, and finally, those features were fed into an ELM. Zhu et al. [125] performed CNN to extract high-level features of images, and K-ELM as a classifier instead of the original linear fully connected layer.

### *7.2. Possibility of Other Random Neural Networks based AE*

Random vector functional link network (RVFL) is a famous random neural network proposed by Pao et al. [25, 26, 27]. As shown in Fig. 14, RVFL is also a SLFN, in which the input layer is directly connected to both the hidden layer and the output layer. Its input weights are randomly generated, while the output weights are calculated through Moore-Penrose pseudo inverse. Pao et al. [26] pointed out that not all the weights in RVFL are equally important, so it is not necessary to iteratively tune all of them.

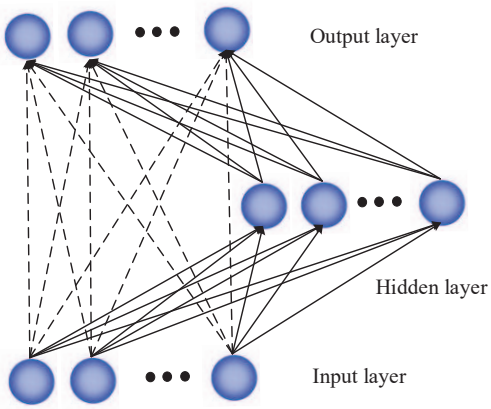



Figure 14: Structure of RVFL

Another typical random neural network was proposed by Schmidt et al. [24]. As shown in Fig. 15, Schmidt's method has no direct link between the input layer and the output layer. Its input weights are randomly selected and kept same throughout the training process, the output weights can be obtained through Fisher method.

As mentioned above, ELM can be used for constructing ELM-AE, and easily extended to hierarchical structure for representation learning. However, when RFVL is implemented for constructing AE, the weights of the direct link connecting the input layer and the output layer will be a constant value one, and the weights of the links between its hidden layer to the output layer will be a constant value zero. Therefore, RFVL will lose the learning capability in AE cases. Schmidt's method may also face difficulty in AE cases due to the biases in the output nodes.

**Remark 7.1:** ELM has been successfully implemented in several applications with very fast learning speed and good generalization performance. For example, ELMs usually can achieve satisfactory performance in dealing with noisy data, indicating the relatively excellent robustness [126, 127, 128, 129, 130, 131]. ELM-AE was proposed based on ELM, thus it naturally inherits the characteristics and capabilities of ELM.

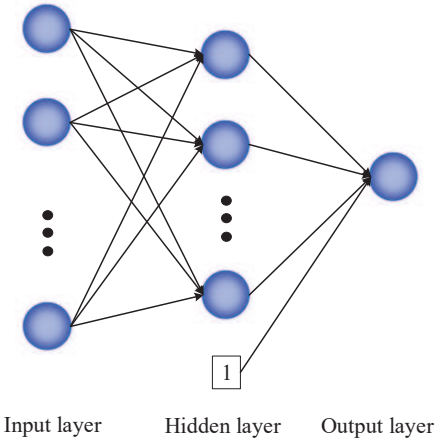



Figure 15: Structure of Schmidt's Method

### *7.3. Conventional Deep Learning with Randomization*

RBM is a probabilistic graphical model based on stochastic neural network with two layers, including a visible layer and a hidden layer. Nodes in the visible layer are fully connected to the nodes in the hidden layer, while there are no connections between nodes in the same layer. Rosa et al. [132] merged RBM and randomized algorithms, and proposed a deep structure for nonlinear system identification, in which the distributions of the hidden weights were trained utilizing the input data and RBM. It shows that combining RBM and randomized algorithms can perform better for nonlinear system identification. Zhang et al. [133] proposed an incremental ELM based on deep feature embedded (IELM-DFE) approach. In the proposed IELM-DFE, the visible layer units are fully connected, and there are two hidden layers for feature extraction, in which the first hidden and visible layer are constructed as a semi-restricted Boltzmann machine (SRBM) model, and the second layer is also used as the hidden layer of ELM. Accordingly, a SRBM model and a RBM model are included in IELM-DFE, the SRBM model mainly for feature extraction, the second RBM model for providing feature expression, and the output weights are calculated through manifold regularization ELM. Wang et al. [3] proposed a noniterative way to train multilayer feedforward neural networks, including two components: an initial extractor of features based on RBM, and a solution of a system of linear matrix equations. In the proposed method, iterative tuning of parameters is not necessary, it is essentially a modified version of random weights assignment based training method by replacing the randomly assigned weights with RBM-based initial weights while keeping the output weights acquired analytically. In addition, for the tasks with strong temporal dependencies among subsequent patterns, recurrent neural network (RNN) usually can achieve satisfactory performance. However, the training of fully adaptable RNN requires the flow of errors gradient information throughout temporal instants, increasing the likelihood of vanishing or exploding gradients and may lead to unstable network behaviors. Actually, for the tasks that do not require a relatively long memory of its inputs, an alternative to a fully adaptable RNN is achieved by introducing the random weight assignment mechanism, allowing for a recurrent layer of fixed, randomly generated nonlinearities, and followed by an adaptable linear layer in the output [134].

Since LeNet-5 was proposed [135], CNN-based deep neural networks, such as AlexNet [136] and VGG [137], can efficiently handle image processing problems due to the special mechanism, including shared weights, sub-sampling, and local receptive fields, etc. Jarret et al. [138] found that random filters used in the two-stage feature extraction system could obtain comparative performance with the approach using pre-training and exact fine tuning of the filters. It means that the network structure is more important than hidden layer parameters in deep learning, indicating that not all the hidden layer parameters need the fine-tuned if we can construct a good hierarchical structure [139]. Zhang et al. [140] proposed the convolutional random vector functional link (CRVFL) by combining RVFL and CNN, in which the convolutional filters were randomly initialized and kept same, only the parameters in the fully connected layers needed to be learned. Compared with classic CNN and its variants, the global fine-tuning is not necessary in CRVFL, and it is not sensitive to the hyper-parameters, such as learning rate, epochs, etc. Wang et al. [141] proposed a convolutional AE ELM (CAE-ELM) combining the advantages of CNN, AE, and ELM. They further designed a modified structure based on the proposed CAE-ELM, which accepted two types of 3D shape representation, i.e.,

voxel data and signed distance field data, as inputs to extract the global and local features of 3D shapes. It is fair to mention that although CNN suffers from heavy computational workloads, it also provides better performance than that of ELMs in image processing tasks. For example, CNN with CIFAR10 dataset provides more than 94% accuracy, while ELM-based neural network only achieves up to 80%. Therefore, how to fully embed convolutional nodes in the ELM-based framework may be a key for performance improvement.

**Remark 7.2:** The essence of combining RBM and random neural network is that replacing the random weight assignment with RBM-based weight initialization and keeping the weights if output layer nodes are calculated analytically. This kind of non-iterative way can guarantee the computational efficiency of RBM-based deep neural network.

## 8. Conclusions

In this paper, we have presented a thorough review on the development of ML-ELM. Some widely-used hierarchical structures are investigated, and selected applications on pattern recognition, image classification, and computer vision are highlighted. Specifically, three typical ML-ELMs, including stacked ELM-AE, residual ELM, and ELM-LRF, are discussed in detail. In addition, we also analyze whether other random neural networks can be used for constructing AE, and the randomization in conventional deep learning approaches. ML-ELM makes deep learning non-iterative and faster due to its random feature mapping mechanism. In addition, the combination of ELM and conventional deep learning approaches can significantly guarantee the computational efficiency of deep learning.

The followings are the challenging issues of ML-ELM, which may accelerate the development of ELM-based deep learning:

1) Impacts of distribution form for randomly generating hidden layer parameters. The randomly generated hidden layer parameters enable ELM training very fast, but it also may lead to the instability. ELM and its variants are sensitive to the randomization range sometimes, whose change may seriously degrade the performance. However, there is still no appropriate criterion to set the randomization range for different tasks depending on the data distributions.

2) Theoretically justifying the effectiveness of random feature mapping in ML-ELM. As mentioned above, random feature mapping ensures the universal approximation and classification approximation of ELM, making its learning efficient with good generalization performance. However, there is still no rigorously theoretical proof for the effectiveness of random feature mapping in ML-ELM. We conjecture that it will be helpful for investing the connection between ML-ELM and other related deep learning approaches.

3) Some conventional deep learning approaches, such as CNN, DBM and DBN, usually require large number of data and tuned parameters to guarantee their excellent performance. Whether the combination of conventional deep learning approaches and ML-ELM can significantly reduce the scale of tuned parameters without performance loss urgently needs to be investigated.


## Acknowledgment

This work is supported in part by the China Postdoctoral Science Foundation under Grants 2019TQ0002 and 2019M660328, the National Natural Science Foundation of China under Grant 61673055, and the National Key Research and Development Program of China under Grant 2017YFB1401203.


## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References


[1] A. Prieto, B. Prieto, E.M. Ortigosa, E. Ros, F. Pelayo, J. Ortega, I. Rojas, Neural networks: an overview of early research, current frameworks and new challenges, Neurocomptuing 214 (2014) 242-268.

[2] Y. LeCun, Y. Bengio, G.E. Hinton, Deep learning, Nature 521 (2015) 436-444.

[3] X. Wang, T. Zhang, R. Wang, Noniterative deep learning: incorporating restricted Boltzmann machine into multilayer random weight neural networks, IEEE Trans. Syst. Man Cybern. Syst. 49 (7) (2019) 1299-1308.

[4] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504-507.

[5] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527-1554.
[6] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798-1828.
[7] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomptuing 70 (2006) 489-501.
[8] J. Cao, J. Hao, X. Lai, C.M. Vong, M. Luo, Ensemble extreme learning machine and sparse representation classification, J. Franklin Inst. 353 (2016) 4526-4541.
[9] P.K. Wong, X. Gao, K.I. Wong, C.M. Vong, Online extreme learning machine based modeling and optimization for point-by-point engine calibration, Neurocomptuing 277 (2018) 187-197.
[10] C.M. Vong, J. Du, C.M. Wong, J. Cao, Postboosting using extended G-mean for online sequential multiclass imbalance learning, IEEE Trans. Neural Netw. Lear. Syst. 29 (12) (2018) 6163-6177.
[11] J. Du, C.M. Vong, Robust online multilabel learning under dynamic changes in data distribution with labels, IEEE Trans. Cybern. 50 (1) (2019) 374-385.
[12] C. Chen, Y. Gan, C.M. Vong, Extreme semi-supervised learning for multiclass classification, Neurocomputing 376 (2020) 103-118.
[13] J. Luo, C.M. Vong, P.K. Wong, Sparse Bayesian extreme learning machine for multi-classification, IEEE Trans. Neural Netw. Lear. Syst. 25 (4) (2014) 836-843.
[14] G.B. Huang, D. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cyber. 2 (2011) 107-122.
[15] G. Huang, G.B. Huang, S. Song, K. You, Trends in extreme learning machines: a review, Neural Netw. 61 (2015) 32-48.
[16] C. Deng, G.B. Huang, J. Xu, J. Tang, Extreme learning machines: new trends and applications, Sci. China Inform. Sci. 58 (2015) 1-16.
[17] S. Ding, H. Zhao, Y. Zhang, X. Xu, R. Nie, Extreme learning machine: algorithm, theory and applications, Artif. Intell. Rev. 44 (2015) 103-115.
[18] G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern. Part B Cybern. 42 (2) (2012) 513-529.
[19] G.B. Huang, M. Li, L. Chen, C.K. Siew, Incremental extreme learning machine with fully complex hidden nodes, Neurocomptuing 71 (2008) 576-583.
[20] G.B. Huang, L. Chen. Convex incremental extreme learning machine, Neurocomptuing 70(2007) 3056-3062.
[21] G.B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomptuing 71 (2008) 3460-3468.
[22] G.B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879-892.
[23] D.S. Broomhead, D. Lowe, Multi-variable functional interpolation and adaptive networks, Complex Syst. 2 (1988) 321-355.
[24] W.F. Schmidt, M.A. Kraaijveld, R.P.W. Dum, Feedforward neural networks with random weights, In: Proceedings of the 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems, 1992, pp. 1097-1105.
[25] Y.H. Pao, Y. Takefuji, Functional-link net computing: theory, system, architecture, and functionalities, Computer 25 (5) (1992) 76-79.
[26] Y.H. Pao, G.H. Park, D.J. Sobajic, Learning and generalization charactertics of the random vector functional-link net, Neurocomptuing 6 (2) (1994) 163-180.
[27] B. Igelnik, Y.H. Pao, Stochastic choice of basis functions in adaptive function approximation and the functional-link net, IEEE Trans. Neural Netw. 6 (6) (1995) 1320-1329.
[28] G.B. Huang, An insight into extreme learning machines: random neurons, random features and kernels, Cogn. Comput. 6 (2014) 376-390.
[29] G.B. Huang, What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle, Cogn. Comput. 7 (2015) 263-278.
[30] W. Zong, G.B. Huang, Y. Chen, Weighted extreme learning machine for imbalance learning, Neurocomptuing, 101 (2013) 229-242.
[31] W. Xiao, J. Zhang, Y. Li, S. Zhang, W. Yang, Class-specific cost regulation extreme learning machine for imblanced classification, Neurocomputing 261 (2017) 70-82.
[32] Y. Li, S. Zhang, Y. Yin, W. Xiao, J. Zhang, Parallel one-class extreme learning machine for imbalance learning based on Bayesian approach, J. Amb. Intel. Hum. Comp. (2018) https://doi.org/10.1007/s12652-018-0994-x.
[33] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3642-3649.
[34] F. Agostinelli, M.R. Anderson, H. Lee, Adaptive multi-column deep neural networks with application to robust image denoising, Adv. Neural Inf. Process. Syst. 26 (2013) 1493-1501.
[35] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecgnizable images, In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 427-436.
[36] N. Morgan, Deep and wide: multiple layers in automatic speech recognition, IEEE Trans. Audio, Speech, Lang. Process. 20 (1) (2012) 7-13.
[37] S. Xue, O.Abdel-Hamid, H. Jiang, L. Dai, Q. Liu, Fast adaptation of deep neural network based on discriminant codes for speech recognition, IEEE/ACM Trans. Audio, Speech, Lang. Process. 22 (12) (2014) 1713-1725.
[38] T.N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A.R. Mohamed, G. Dahl, B. Ramabhadran, Deep convolutional neural networks for large-scale speech tasks, Neural Netw. 64 (2015) 39-48.
[39] J. Gonzalez-Donminguez, I. Lopez-Moreno, P.J. Moreno, J. Gonzalez-Rodriguez, Frame-by-frame language identification in short utterances using deep neural networks, Neural Netw. 64 (2015) 49-58.
[40] M. Sun, X. Zhang, T. Zheng, Unseen noise estimation using separable deep autoencoder for speech recognition, IEEE/ACM Trans. Audio, Speech, Lang. Process. 24 (1) (2016) 93-104.
[41] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, Adv. Neural Inf. Process. Syst. 19 (2007) 153-160.
[42] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85-117.
[43] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11-26.
[44] D.H. Ballard, Modular learning in neural networks, In: Proceedings of the 6th National Conference on Artificial Intelligence, 1987, pp.

279-284.
[45] H. Bourlard, P. Lamblin, Auto-association by multilayer perceptrons and singular value decomposition, Bio. Cybern. 59 (4-5) (1988) 291-294.
[46] L.L.C. Kasun, H. Zhou, G.B. Huang, C.M. Vong, Representational learning with ELMs for big data, IEEE Intell. Syst. 28 (6) (2013) 31-34.
[47] H. Cecotti, Deep random vector functional link network for handwritten character recognition, In: Proceedings of the 2016 International Conference on Neural Networks, 2016, pp. 3628-3633.
[48] C.M. Wong, C.M. Vong, P.K. Wong, J. Cao, Kernel-based multilayer extreme learning machines for representation learning, IEEE Trans. Neural Netw. Lear. Syst. 29 (3) (2018) 757-762.
[49] C.M. Vong, C. Chen, P.K. Wong, Empirical kernel map-based multilayer extreme learning machines for representation learning, Neurocomptuing, 310 (2018) 265-276.
[50] J. Tang, C. Deng, G.B. Huang, Extreme learning machine for multilayer perceptron, IEEE Trans. Neural Netw. Lear. Syst. 27 (4) (2016) 809-821.
[51] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imag. Sci. 2 (1) (2009) 183-202.
[52] A. Beck, M. Teboulle, Fast grandient-based algorithms for constrained total variation image denoising and deblurring problems, IEEE Trans. Image Process. 18 (11) (2009) 2419-2434.
[53] L. Chen, H. Paul, H. Qu, J. Zhao, X. Sun, Correntropy-based robust multilayer extreme learning machines, Pattern Recog. 84 (2018) 357-370.
[54] H. Zhou, G.B. Huang, Z. Lin, H. Wang, Y.C. Soh, Stacked extreme learning machines, IEEE Trans. Cybern. 45 (9) (2015) 2013-2025.
[55] X. Luo, Y. Xu, W. Wang, M. Yuan, X. Ban, Y. Zhu, W. Zhao, Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy, J. Franklin Inst. 355 (2018) 1945-1966.
[56] H. Dai, J. Cao, T. Wang, M. Deng, Z. Yang, Multilayer one-class extreme learning machine, Neural Netw. 115 (2019) 11-22.
[57] J. Zhang, W. Xiao, Y. Li, S. Zhang, Z. Zhang, Multilayer probability extreme learning machine for device-free localization, Neurocomputing (2019) https://doi.org/10.1016/j.neucom.2018.11.106.
[58] P. Vincent, H. Larochelle, Y. Bengio, P.A. Manzagol, Extracting and composing robust features with denoising autoencoders, In: Proceedings of the 25th International Conference on Machine Leaning, 2008, pp. 1096-1103.
[59] H. Larochelle, Y. Bengio, J. Louradour, P. Lamblin, Exploring strategies for training deep neural networks, J. Mach. Learn. Res. 1 (2009) 1-40.
[60] N. Zhang, S. Ding, Z. Shi, Denoising Laplacian multi-layer extreme learning machine, Neurocomputing 171 (2016) 1066-1074.
[61] L. Cao, W. Huang, F. Sun, Building feature space of extreme learning machine with sparse denoising stacked-autoencoder, Neurocomputing 174 (2016) 60-71.
[62] J. Hu, J. Zhang, C. Zhang, J. Wang, A new deep neural network based on a stack of single-hidden-layer feedforward neural networks with randomly fixed hidden neurons, Neurocomputing 171 (2016) 63-72.
[63] G. Huang, S. Song, J.N.D. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Trans. Cybern. 44 (12) (2014) 2405-2417.
[64] K. Sun, J. Zhang, C. Zhang, J. Hu, Generalized extreme learning machines autoencoder and a new deep neural network, Neurocomputing 230 (2017) 374-381.
[65] Y. Gu, Y. Chen, J. Liu, X. Jiang, Semi-supervised deep extreme learning machine for Wi-Fi based localization, Neurocomputing 166 (2015) 282-293.
[66] C. Chen, K. Li, A. Ouyang, Z. Tang, K. Li, GPU-accelerated parallel hierarchical extreme learning machine on Flink for big data, IEEE Trans. Syst. Man Cybern. Syst. 47 (10) (2017) 2740-2753.
[67] L. Yao, Z. Ge, Distributed parallel deep learning of hierarchical extreme learning machine for multimode quality prediction with big process data, Eng. Appl. Artif. Intel. 81 (2019) 450-465.
[68] M.D. Tissera, M.D. McDonnel, Deep extreme learning machines: supervised autoencoding architecture for classification, Neurocomputing 174 (2016) 42-49.
[69] X. Wen, H. Liu, G. Yan, F. Sun, Weakly paired multimodal fusion using multilayer extreme learning machine, Soft Comput. 22 (2018) 3533-3544.
[70] Y. Chu, C. Feng, C. Guo, Y. Wang, Network embedding based on deep extreme learning machine, Int. J. Mach. Learn. Cybern. (2018) https://doi.org/10.1007/s13042-018-0895-5.
[71] Y. Yang, Q.M.J. Wu, Y. Wang, Autoencoder with invertible functions for dimension reduction and image reconstruction, IEEE Trans. Syst. Man Cybern. Syst. 48 (7) (2018) 1065-1079.
[72] Y. Yang, Q.M.J. Wu, Multilayer extreme learning machine with subnetwork nodes for representation learning, IEEE Trans. Cybern. 46 (11) (2016) 2570-2583.
[73] X. Wang, R. Wang, C. Xu, Discovering the relationship between generalization and uncertainty by incorporating complexity of classification, IEEE Trans. Cybern. 48 (2) (2018) 703-715.
[74] W. Cao, X. Wang, Z. Ming, J. Gao, A review on neural networks with random weights, Neurocomputing 275 (2018) 278-287.
[75] Y. Yang, Q.M.J. Wu, Extreme learning machine with subnetwork hidden nodes for regression and classification, IEEE Trans. Cybern. 46 (12) (2016) 2885-2898.
[76] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, S. Mao, G.B. Huang, A theoretical study of the relationship between an ELM network and its subnetworks, In: Proceedings of 2017 International Joint Conference on Neural Networks, 2017, pp. 1794-1801.
[77] Y. Yang, Q.M.J. Wu, W.L. Zheng, B.L. Lu, EEG-based emotion recognition using hierarchical network with subnetwork nodes, IEEE Trans. Cong. Dev. Syst. 10 (2) (2018) 408-419.
[78] W. Wu, Q.M.J. Wu, W. Sun, Y. Yang, , X. Yuan, W.L. Zheng, B.L. Lu, A regression method with subnetwork neurons for vigilance estimation using EOG and EEG, IEEE Trans. Cong. Dev. Syst. (2018) DOI: 10.1109/TCDS.2018.2889223.
[79] Y. Yang, Q.M.J. Wu, Features combined from hundreds of midlayers: hierarchical networks with subnetwork nodes, IEEE Trans. Neural Netw. Lear. Syst. 30 (11) (2019) 3313-3325.
[80] B. Mriza, S. Kok, F. Dong, Multi-layer online sequential extreme learning for image classification, In: Proceedings of ELM-2015, 2016, pp. 39-49.

[81] X. Su, S. Zhang, Y. Yin, Y. Hui, W. Xiao, Prediction of hot metal silicon content for blast furnace based on multi-layer online sequential extreme learning machine, In: Proceedings of the 37th Chinese Control Conference, 2018, pp. 8025-8030.
[82] Y. Li, S. Zhang, Y. Yin, W. Xiao, J. Zhang, A novel online sequential extreme learning machine for gas utilization ratio prediction in blast furnaces, Sensors 17 (8) (2017) 1847-1870.
[83] N. Liang, G.B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online squential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17 (6) (2006) 1411-1423.
[84] S. Scardapane, D. Comminiello, M. Scarpiniti, A. Uncini, Online sequential extreme learning machine with kernels, IEEE Trans. Neural Netw. Lear. Syst. 26 (9) (2015) 2214-2220.
[85] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.
[86] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, In: Proceedings of the 14th European confernece on computer vision, 2016, pp. 1-15.
[87] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, In: Proceedings of the 31st AAAI confernece on artificial intelligence, 2017, pp. 1-7.
[88] A. Veit, M. Wilber, S. Belongie, Residual networks behave like ensemble of relatively shallow networks, In: Proceedings of the advance in neural information processing systems, 2016, pp. 550-558.
[89] J. Zhang, W. Xiao, Y. Li, S. Zhang, Residual compensation extreme learning machine for regression, Neurocomputing 311 (2018) 126-136.
[90] J. Zhang, Y. Lu, B. Zhang, W. Xiao, Device-free localization using empirical wavelet transform-based extreme learning machine, In: Proceedings of the 30 Chinese Conference on Control and Decision, 2018, pp. 2585-2590.
[91] M.D. Tissera, M.D. McDonnell, Modular expansion of the hidden layer in single layer feedforward neural networks, In: Proceedings of 2016 International Joint Conference on Neural Networks, 2016, pp. 2939-2945.
[92] W. Yu, F. Zhang, Q. He, Z. Shi, Learning deep representations via extreme learning machines, Neurocomputing 149 (2015) 308-315.
[93] T. Wang, J. Cao, X. Lai, B. Chen, Deep weighted extreme learning machine, Cognit. Comput. 10 (6) (2018) 890-907.
[94] Y. Yang, Q.M.J. Wu, Y. Wang, K.M. Zeeshan, X. Lin, X. Yuan, Data partition learning with multiple extreme learning machines, IEEE Trans. Cybern. 45 (8) (2018) 1463-1475.
[95] G.B. Huang, Z. Bai, L.L.C. Kasun, C.M. Wong, Local receptive fields based extreme learning machine, IEEE Comput. Intell. Mag. 10 (2) (2015) 18-29.
[96] Z. Bai, L.L.C. Kasun, G.B. Huang, Generic object recognition with local receptive fields based extreme learning machines, Procedia Comput. Sci. 53 (2015) 391-399.
[97] H. Liu. F. Li, X. Xu, F. Sun, Multi-modal local receptive field extreme learning machine for object recognition, Neurocomputing 277 (2018) 4-11.
[98] H. Liu. F. Li, X. Xu, F. Sun, Active object recognition using hierarchical local-receptive-field-based extreme learning machine, Memetic Comp. 10 (2018) 233-241.
[99] F. Li, H. Liu. X. Xu, F. Sun, Haptic recognition using hierarchical extreme learning machine with local-receptive-field, Int. J. Mach. Learn. Cybern. (2019) https://doi.org/10.1007/s13042-017-0736-y.
[100] H. Li, H. Zhao, H. Li, Neural-response-based extreme learning machine for image classification, IEEE Trans. Neural Netw. Lear. Syst. 30 (2) (2019) 539-552.
[101] W. Zhu, J. Miao, L. Qing, G.B. Huang, Hierarchical extreme learning machine for unsupervised representation learning, In: Proceedings of 2015 International Joint Conference on Neural Networks, 2015, pp. 1-8.
[102] W. Wang, C.M. Vong, Y. Yang, P.K. Wong, Encrypted image classification based on multilayer extreme learning machine, Multidim. Syst. Sign. Process. 28 (2017) 851-865.
[103] M. Ahmad, A.M. Khan, M. Mazzara, S. Distefano, Multi-layer extreme learning machine-based autoencoder for hyperspectral image classification, In: Proceedings of the 14th International Conference on Computer Vision Theory and Applications, 2019, pp. 1-8.
[104] F. Cao, Z. Yang, J. Ren, W. Chen, G. Han, Y. Shen, Local block multilayer sparse extreme learning machine for effective feature extraction and classification of hyperspectral images, IEEE Trans. Geosci. Remote Sens. 57 (8) (2019) 5580-5594.
[105] D.R. Nayak, D. Das, R. Dash, S. Majhi, B. Majhi, Deep extreme learning machine with leaky rectified linear unit for multiclass classification of pathological brain images, Multidim. Tools Appl. (2019) https://doi.org/10.1007/s11042-019-7233-0.
[106] Y. Yang, H. Zhang, D. Yuan, D. Sun, G. Li, R. Ranjan, M. Sun, Hierarchical extreme learning machine based image denoising network for visual Internet of Things, Appl. Soft Comput. 74 (2019) 747-759.
[107] H. Yu, J. Wang, X. Sun, Surveillance video online prediction using multilayer ELM with object principal trajectory, Signal Image. Video Process. (2019) https://doi.org/10.1007/s11760-019-01471-y.
[108] L. Duan, M. Bao, J. Mao, Y. Xu, J. Chen, Classification based on multilayer extreme learning machine for motor imagery task from EEG signals, Procedia Comput. Sci. 88 (2016) 186-184.
[109] Q. She, B. Hu, Z. Luo, T. Nguyen, Y. Zhang, A hierarchical semi-supervised extreme learning machine method for EEG recognition, Med. Biol. Eng. Comput. 57 (1) (2019) 147-157.
[110] S.T. Kadam, V.M.N. Dhaimodker, M.M. Patil, D.R. Edla, V. Kuppili, EIQ: EEG based IQ test using wavelet packed transform and hierarchical extreme learning machine, J. Neurosci. Meth. 322 (2019) 71-82.
[111] Z. Yin, J. Zhang, Task-generic mental fatigue recognition based neurophysiological signal and dynamic deep extreme learning machine, Neurocomputing 283 (2018) 266-281.
[112] S. Ding, N. Zhang, X. Xu, L. Guo, J. Zhang, Deep extreme learning machine and its application in EEG classification, Math. Probl. Eng. 835 (2015) Article ID 129021, 11 pages.
[113] X. Niu, Z. Wang, Z. Pan, Extreme learning machine based deep model for human activity recognition with wearable sensors, Comput. Syst. Sci. 21 (5) (2019) 16-25.
[114] M. Chen, Y. Li, X. Luo, W. Wang, L. Wang, W. Zhao, A novel human activity recognition scheme for smart health using multilayer extreme learning machine, J. Neurosci. Meth. 6 (2) (2019) 1410-1418.

[115] W. Ibrahim, M.S. Abadeh, Extracting features from protein sequences to improve deep extreme learning machine for protein fold recognition, J. Theor. Biol. 421 (2017) 1-15.
[116] W. Ibrahim, M.S. Abadeh, Protein fold recognition using deep kernelized extreme learning machine and linear discriminant analysis, Neural Comput. Appl. 31 (2019) 4201-4214.
[117] R.K. Roul, S.R. Asthana, G. Kumar, Study on suitability and importance of multilayer extreme learning machine for classification of text data, Soft Comput. 21 (2017) 4239-4256.
[118] R. Cao, J. Cao, J.P. Mei, C. Yin, X. Huang, Radar emitter identification with bispectrum and hierarchical extreme learning machine, Multimed Tools Appl. 78 (20) (2018) 28953-28970.
[119] L. Zhang, H. Yang, Z. Jiang, Imbalanced biomedical data classification using self-adaptive multilayer ELM combined with dynamic GAN, BioMed. Eng. OnLine. 17 (2018) 181-201.
[120] O. Barak, M. Rigotti, S. Fusi, The sparsemess of mixed selectivity neurons controls the generalization-discrimination trade off, J. Neurosci. 33 (9) (2013) 3844-3856.
[121] M. Rigotti, O. Barak, M.R. Warden, X.J. Wang, N.D. Daw, E.X. Miller, S. Fusi, The importance of mixed selectivity in complex cognitive tasks, Nature 497 (2013) 585-590.
[122] S. Fusi, E.K. Miller, M. Rigotti, Why neurons mix: high dimensionality for higher cognition, Curr. Opin. Neurobiol. 37 (2015) 66-74.
[123] B. Ribeiro, N. Lopes, Extreme learning machine with deep concepts, In: Proceedings of the Iberoamerican Congress on Pattern Recognition, 2013, pp. 182-189.
[124] K. Han, D. Yu, I. Tashev, Speech emotion recognition using deep neural network and extreme learning machine, In: Proceedings of the Interspeech, 2014, pp. 223-227.
[125] X. Zhu, Z. Li, X.Y. Zhang, P. Li, Z. Xue, L. Wang, Deep convolutional representations and kernel extreme learning machines for image classification, Multimed. Tools Appl. (2018) https://doi.org/10.1007/s11042-018-6781-z.
[126] J. Cao, Y. Zhao, X. Lai, M.E.H. Ong, C. Yin, Z.X. Koh, N. Liu, Landmark recognition with sparse representation classification and extreme learning machine, J. Franklin Inst. 352 (2015) 4528-4545.
[127] Y. Li, S. Zhang, Y. Yin, J. Zhang, and W. Xiao, A soft sensing scheme of gas utilization prediction for blast furnace via improved extreme learning machine, Neural Process. Lett., 50 (2) (2019) 1191-1213.
[128] I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, E. Cambria, Bayesian network based extreme learning machine for subjectivity detection, J. Franklin Inst. 355 (2018) 1780-1797.
[129] Y. Li, S. Zhang, J. Zhang, Y. Yin, W. Xiao, Z. Zhang, Data-driven multi-objective optimization for burden surface in blast furnace with feedback compensation, IEEE Trans. Ind. Informat. 16 (4) (2020) 2233-2244.
[130] Y. Li, H. Li, J. Zhang, S. Zhang, Y. Yin, Burden surface decision Using MODE with TOPSIS in blast furnace ironmkaing, IEEE Access, 8 (2020) 35712-35725.
[131] J. Zhang, W. Xiao, S. Zhang, and S. Huang, Device-free localization via an ectreme learning machine with paramterized geometrical feature extraction, Sensors, 17 (4) (2017) 879-899.
[132] E. de la Rosa, W. Yu, Randomized algorithms for nonlinear system identification with deep learning modification, Inf. Sci. 365 (2016) 197-212.
[133] J. Zhang, S. Ding, N. Zhang, Z. Shi, Incremental extreme learning machine based on deep feature embedded, Int. J. Mach. Learn. Cybern. 7 (2016) 111-120.
[134] S. Scardapane, D. Wang, Randomness in neural networks: an overview, WIREs. Data Min. Knowl. 7 (2) (2018) 1-18.
[135] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1988) 2278-2324.
[136] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, In: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1097-1105.
[137] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, In: Proceedings of the 2015 International Conference on Learning Representations, 2015, pp. 1-14.
[138] K. Jarrett, K. Kavukcuoglu, M.A. Ranzato, Y. LeCun, What is the best multi-stage architecture for object recognition? In: Proceedings of IEEE 12th International Conference on Computer Vision, 2009, pp. 2146-2153.
[139] A. Saxe, P.W. Koh, Z. Chen, M. Bhand, B. Suresh, A.Y. Ng, On random weights and unsupervised feature learning, In: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 1089-1096.
[140] L. Zhang, P.N. Suganthan, Visual tracking with convolutional random vector functional link network, IEEE Trans. Cybern. 47 (10) (2017) 3243-3253.
[141] Y. Wang, Z. Xie, K. Xu, Y. Dou, Y. Lei, An efficient and effective convolutional auto-encoder extreme learning machine network for 3D feature learning, Neurocomputing 174 (2016) 988-998.