Neurocomputing 000 (2017) 1–10

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A review on neural networks with random weights

Weipeng Cao[a], Xizhao Wang[a,*], Zhong Ming[a], Jinzhu Gao[b]

[a] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[b] School of Engineering and Computer Science, University of the Pacific, CA 95211, USA

ARTICLE INFO

Article history:
Received 31 May 2017
Revised 8 July 2017
Accepted 14 August 2017
Available online xxx

Communicated by Guan Ziyu.

Keywords:
Feed-forward neural networks
Training mechanism
Neural networks with random weights

ABSTRACT

In big data fields, with increasing computing capability, artificial neural networks have shown great strength in solving data classification and regression problems. The traditional training of neural networks depends generally on the error back propagation method to iteratively tune all the parameters. When the number of hidden layers increases, this kind of training has many problems such as slow convergence, time consuming, and local minima. To avoid these problems, neural networks with random weights (NNRW) are proposed in which the weights between the hidden layer and input layer are randomly selected and the weights between the output layer and hidden layer are obtained analytically. Researchers have shown that NNRW has much lower training complexity in comparison with the traditional training of feed-forward neural networks. This paper objectively reviews the advantages and disadvantages of NNRW model, tries to reveal the essence of NNRW, gives our comments and remarks on NNRW, and provides some useful guidelines for users to choose a mechanism to train a feed-forward neural network.

1. Introduction

Artificial neural networks (ANNs) have received considerable attention due to its powerful ability in image processing, speech recognition, natural language processing, etc. The ANN models performance depends largely on the quantity and quality of data, computing power, and the efficiency of algorithms. Traditional ANNs train models by iteratively tuning all the weights and biases in minimizing a loss function which is defined as the difference between model predictions and real observations. During the training process, the derivatives of the loss function are back propagated to each layer to guide parameter adjustment [1]. Unfortunately this method has several critical drawbacks, such as slow convergence, local minima problem, and model selection uncertainty.

Deep learning refers to train a multilayer neural network by using a gradient based technique, which has become an unprecedented hot research topic after AlphaGo, an artificial intelligence program based on deep learning technology, beat Lee Sedol, the famous 18-time Go world champion [2]. Deep learning trains models in a similar way as the traditional ANNs do. In deep learning, all the parameters are first initialized by using unsupervised methods and then are tuned by using Back Propagation (BP) technique method [1]. The multilayer architecture can be treated as a whole and all the internal parameters need to be fine-tuned iteratively. As the depth increases, training a deep learning model needs tremendous amount of time, even on the powerful GPU-based computers [3–7]. In addition, deep learning with BP has all weaknesses that ANNs have.

Neural network with random weights (NNRW) provides a solution for the problems that traditional ANNs and the BP-based deep learning approaches have. NNRW is defined as a non-iterative training algorithm in which the hidden weights and biases are randomly selected from a given range and kept same throughout the training process while the weights between the hidden layer and the output layer are obtained analytically. Compared with traditional learning with global tuning such as deep learning with BP-based method, NNRW can achieve much faster training speed with acceptable accuracy. In addition, NNRW is easy to implement and its universal approximation capability has been proven in theory [8–10].

In recent years, there are several review articles about NNRW have been published. Deng et al. [11] provided an overview of extreme learning machine (ELM) theory and its variants, especially on online sequential ELM (OS-ELM), incremental ELM (I-ELM), ELM ensembles, etc. In addition, [11] mentioned some of the embryos of deep ELM architecture, such as ELM Auto-encoder (ELM-AE) and Multilayer ELM (ML-ELM). With the rapid development of ELM, many improved algorithms and diverse applications have emerged recently. Huang et al. [12] has shown that, apart from

* Corresponding author.
E-mail addresses: xizhaowang@ieee.org, xzwang@szu.edu.cn (X. Wang).

http://dx.doi.org/10.1016/j.neucom.2017.08.040
0925-2312/

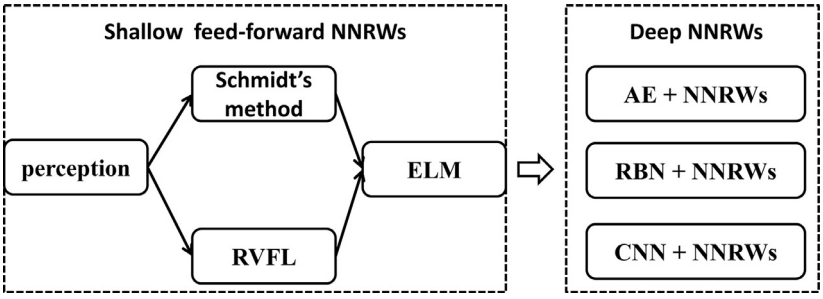

Fig. 1. The evolvement of feed-forward NNRW.

classification and regression, ELM can be extended to deal with compression, feature learning, and clustering. The ELM hardware implementation and parallel computation techniques of ELM are also being mentioned in [12].

Although [11,12] have offered a thorough overview of earlier ELM theory and its applications, there are still two important problems remain untouched. First, the essential idea and training mechanism of ELM are the same as other types of NNRW, such as random vector functional link networks (RVFL) [13] and Schmidts method [14] (refer to Section 2 for details). However, the above review articles do not touch on the development of this field and lack of discussion of these similar algorithms. Second, it is well known that deep neural network can obtain high-level representation from data, which is the core reason for the success of deep learning. Intuitively, NNRW with deep architecture can greatly improve the performance of existing models and be applied to more complex tasks. Actually, a large number of successful cases have sprung up in recent years. However, the above documents lack the relevant contents.

Zhang et al. [15] and Li et al. [16] have conducted a comprehensive study on the relationship between the parameters and the performance of RVFL (which is also a type of NNRW) and given a series of guidance for building RVFL models. Zhang and Suganthan [17] and Scardapane and Wang [18] presented a survey on the evolution of NNRW and its related topics, especially on RVFL, radial basis function network with random weights, and recurrent networks with random weights. However, none of these reviews mention ELM theory and its applications, and there is a lack of comments on the relationship between ELM and other NNRW.

Based on the above reasons, this paper makes a comprehensive survey on the development of NNRW theory and its applications, especially on the discussion of the differences between these algorithms (i.e., RVFL, Schmidts method, ELM, etc.) and the evolution of deep NNRW (such as Auto-encoder with NNRW, restricted Boltzmann machine with NNRW, and convolutional neural networks with NNRW).

It is worth noting that, regarding the focuses and contents, our review is quite different from some existing surveys mentioned above. The architecture of this paper is shown in Fig. 1.

The rest of the paper is organized as follows. We first review the advantages and disadvantages of shallow neural networks with random weights by incorporating our comments in Section 2, and then describe the development of deep neural network with random weights and its applications in Section 3, and finally conclude the paper in Section 4.

2. Shallow feed-forward neural network with random weights and applications

Randomness has been introduced into artificial neural network since the period of perception model [19]. Inspired by the biological nervous system, Rosenblatt [19] designed a perception model, which includes retina of sensory units (S-points, used to transmit stimulation signals to association cells), a set of association cells (A-units, generating reflection of excitement or inhibition according to the value of impulse intensities), and response units (R-unit, giving responses in a similar way as A-units do). In this perception model, the connections between any two A-units are assumed to be random. Rosenblatt observed that this system with randomly connected units can give specific responses to specific stimuli under certain constraints.

Inspired by the pioneers work, Pao et al. [13] proposed Random Vector Functional Link Networks (RVFL) and Schmidt et al. [14] proposed another single layer feed-forward neural network with random weights, both in 1992. Since then, many researchers have devoted tremendous efforts to developing theories and applications for neural networks with random weights. In this section, we will study two typical shallow feed-forward neural networks with random weights and their applications.

2.1. Random Vector Functional Link Networks (RVFL) and its applications

Random Vector Functional Link Networks (RVFL) was proposed by Pao et al. in the 1990s [8,9,13,20,21]. The structure of RVFL is shown in Fig. 2. RVFL is a special single layer feed-forward neural network (SLFN), in which the input layer is directly connected to both the hidden layer and the output layer. The weights between the input layer and hidden layer are randomly selected from $[-1, 1]$, while the weights between the input layer and output layer and the weights between the hidden layer and output layer are obtained by MooreCPenrose pseudo-inverse. The authors pointed out that not all the weights in RVFL are equally important and it is not necessary to iteratively tune all of them [21]. In addition, the authors showed several advantages of RVFL, such as easy hardware implementation [13], fast convergence [21], powerful approximation capability [8,9], and satisfying the requirements of real-time applications [20].

Zhang et al. [15] and Li et al. [16] have conducted a lot of experiments to study the relationship between hidden parameters and the performance of RVFL. Zhang and Suganthan [15] show that the direct links between input layer and output layer have significant impact on the performance of RVFL. Using *Radbas* as activation function often can achieve better performance than using either *hardlim* or sign as activation function. Li and Wang [16] have studied the relationship between the scope setting of hidden parameters and the model performance. They mentioned that it is improper to select hidden weights and biases from an empirically fixed scope setting (i.e., $[-1, 1]$) for any RVFL model.

So far, single hidden layer RVFL and its variants have been widely used in real-world applications. Some notable applications include time-series data prediction [22], English language handwritten script recognition [23], semi-supervised learning [24], hardware implementation [25], conditional probability densities

Please cite this article as: W. Cao et al., A review on neural networks with random weights, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.040

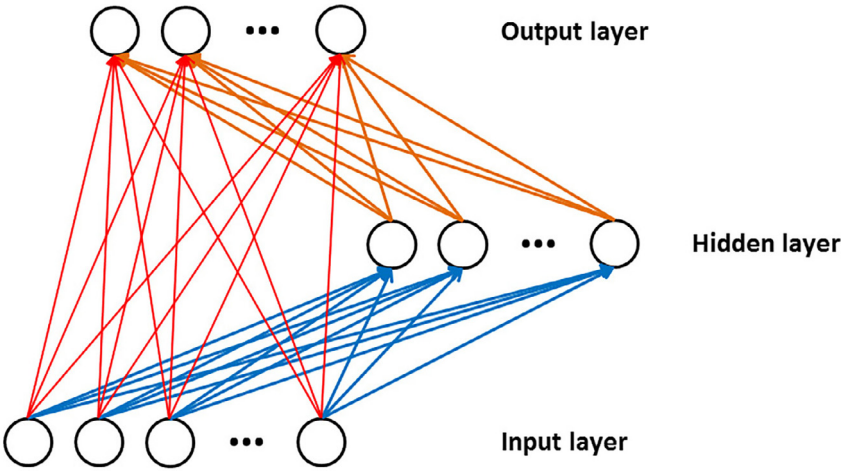

Fig. 2. The structure of RVFL.

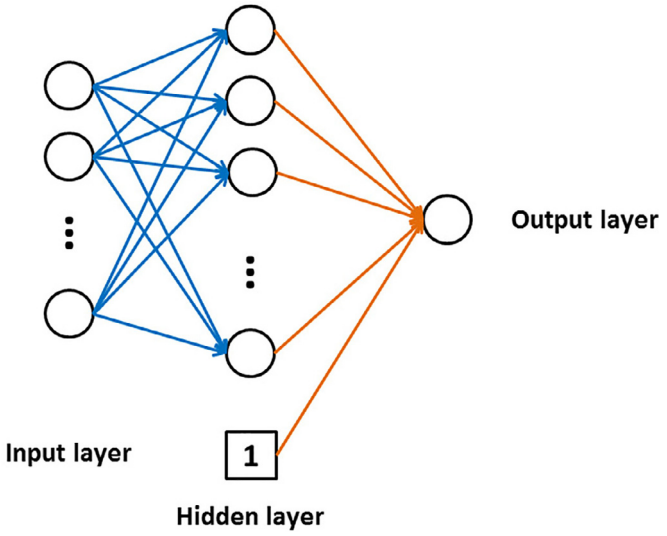

Fig. 3. The structure of Schmidt's method.

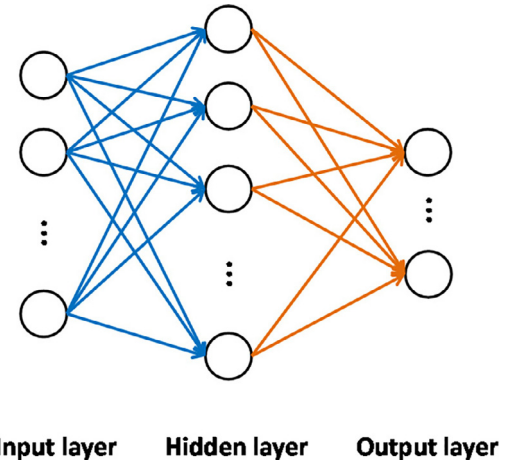

Fig. 4. The structure of ELM.

prediction [26,27], Ensemble learning [28], distributed learning [29], signal enhancement [30], etc.

2.2. Standard feed-forward neural network with random weights and extreme learning machine

At same time when Pao et al. proposed RVFL [8,9,13], Schmidt et al. proposed another standard feed-forward neural network with random weights [14]. The structure of this network is shown in Fig. 3.

Unlike RVFL, there is no direct link between the input layer and output layer. The weights between the input layer and hidden layer are randomly selected and kept same throughout the training process. The weights between the hidden layer and output layer are determined by using the Fisher method. Note that the value multiplied by the node and the weight (between the node and the output node) can be used as the threshold value to absorb the system error. The paper shows that this method can achieve comparable accuracy and much smaller standard deviations compared with the standard back propagation method in low dimensional problems.

Inspired by this work, more and more related theories and applications have been developed in recent years. One of the most attractive theories is proposed by Huang et al. in 2004 with the new name extreme learning machine (ELM) [31]. ELM extends the above standard feed-forward neural network with random weights in many ways, such as setting the bias of the output node to zero, transforming different hidden nodes to one unified form, etc. Ridge regression theory, linear systems stability theory, matrix stability theory, neural network generalization performance theory, and maximal margin theory are also embodied in ELM theory [32–34]. Huang et al. [31] gave a series of theoretical analysis and rigorous theoretical proof for ELM. It shows that almost any nonlinear piecewise continuous random hidden nodes (including Sigmoid nodes, Radial basis function nodes, Wavelet, Fourier series and Biological neurons) can be used in ELM, and the resultant networks have universal approximation capabilities [10]. Huang et al. [35] have proven that support vector machine (SVM, [36]) is a suboptimal solution of ELM and [37] shows that random projection (RP) and principal component analysis (PCA) are special cases of ELM when linear function is used as activation function. In addition, Huang et al. [38] mentioned that ELM theory is inherently consistent with the mechanism of biological learning (a part of neurons are randomly connected and some neurons do not need to be tuned, [39–43]) and the basic learning units in biological learning (compression, feature learning, sparse coding, clustering, and classification) can be implemented by the same ELM architecture. To some extent, ELM is often seen as a generalized feed-forward neural network with random weights [38,44]. The standard network structure of ELM is shown in Fig. 4.

Up to now, the standard feed-forward neural network with random weights [14] and shallow ELMs have been widely applied

Please cite this article as: W. Cao et al., A review on neural networks with random weights, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.040

to many applications. Some notable applications include fuzzy nonlinear regression [45], embedded system [46–48], human face recognition [49–51], human action recognition [52], traffic sign recognition [53], biology and bioinformatics [54,55], image classification [56], indoor localization [57], clustering problem [58,59], high-dimensional data [60], etc. In addition to these typical applications, some improved algorithms are equally worth mentioning. [61] proposed a functional iterative method to optimize the solution of original ELM model and proved this method could converge linearly. Ensemble learning can always achieve better generalization ability [62,63] and [64] showed two different ELM-based ensemble strategies (i.e., ensemble with same architecture and ensemble with different base components) and gave some guidelines for constructing a good ELM-based ensemble model.

2.3. Remarks

This is a special type of methodologies for training feed-forward neural networks, i.e., random assignment of input weights and biases. The following is brief summary historically.

In 1988 Broomhead and Lowe [65] proved implicitly the universal approximation ability of this type of neural networks based on the radial basis function network with random centers.

In 1992 Pao et al. [13] and Schmidts group [14] proposed the Random Vector Functional Link Networks (RVFL) and standard feed-forward neural network with random weights, respectively. The most obvious difference between the two networks is that there is a direct connection between the input layer and output layer in RVFL, while Schmidts method does not.

In 1994 Pao et al. [21] pointed out that it is not necessary to tune all the linking-weights of RVFL iteratively, because most of them are not important. Furthermore, they demonstrated the universal approximation ability of RVFL in [8,9].

In 2006 Huang et al. further investigated this type of neural networks and its training methodology with the new name extreme learning machine (ELM) [10,44].

After 2006 there appear several new names to describe this type of neural networks such as random weight network (RWN) [45,51], neural network with random weights (NNRW) [50], random weight neural network (RNN) [66], feed-forward networks with random weights (RW-FFN) [17], etc.

Essentially the ideas all of above-mentioned neural networks and their training mechanisms are same but some details are slightly different. In other words the similarity is essential while the difference is trivial. We list some differences as follows.

(1) *Network architecture*. The most distinct character of RVFL is that the input layer is directly connected to the output layer, while Schmidts method and ELM do not. ELM has been extended to multiple hidden layers architectures and achieved many outstanding achieves, while RVFL and Schmidts method are mainly used in the single hidden layer cases. In addition, both RVFL and Schmidts method are fully connected networks, while ELM can be both fully connected network and partially connected network. And the hidden nodes in ELM can be sub-networks, while RVFL and Schmidts method can not.
(2) *Hidden node type*. The hidden node type of RVFL is limited to Sigmoid and Radial basis function, Schmidts method only works in Sigmoid cases, while the hidden node type of ELM is extended to Sigmoid, Radial basis function, Wavelet, Fourier series, Biological neurons, etc.
(3) *Training mechanism*. In [65] only the radial basis function centers are randomly selected but not the impact factors, while in ELM both of them are randomly selected. The objective optimization function of ELM is based on both the structural and empirical risks errors, and thus guarantees that the model has better generalization ability, while other methods are not. Compared to ELM, Huang [33] has proved that RVFL and Schmidts method provide suboptimal solutions.
(4) *Universal approximation ability*. The universal approximation ability of ELM and its variations have been rigorously proved in theory, while there is no theoretical proof for Schmidts method. Both Broomhead et al. [65] and Paos group [8,9] also did not give rigorous proof for full random hidden nodes cases.

For convenience in this paper we call this type of neural networks and their training mechanism as neural networks with random weights (NNRW). NNRW, which has a fundamental assumption that not-all-weights are necessarily tuned in the training, indeed overcomes the problems of low training efficiency in comparison with gradient based training, but the following critical issues remain.

(1) *Generalization ability*. There is no sufficient evidence to show that the generalization ability of NNRW is superior to other methods. Although Huang et al. have proved that SVM is a suboptimal solution of ELM [35], however, it is impossible for NNRW to have high prediction accuracy (i.e., the generalization ability) for all the datasets. Up to now, it is still unclear that what kind of problems NNRW can have excellent performance.
(2) *Feedback mechanism*. Owing to the non-iterative learning strategy, NNRW can achieve much faster learning speed. There's no such thing as a free lunch, without the iterative tuning of weights, NNRW may not be able to obtain the semantic meaning of learned weights. The impact of this non-feedback mechanism remains to be investigated. In addition, the random feature mapping is the core idea of NNRW, the effective evaluation of the random feature mapping remains untouched.
(3) *Model stability*. The randomization range and the type of distribution of the hidden parameters have significant impact on the performance of NNRW. Unfortunately, there is still no good way to guide the selection of the hidden parameters. Different random parameters produce different models, and thus cause the model performance to be unstable. How to solve this problem remains to be studied. In addition, the activation function of NNRW is also playing an important role in the model stability. Zhang et al. [15] showed that *Radbas* always achieves better performance while *hardlim* and *sign* always degenerate the performance of RVFL. Huang et al. [12] demonstrated that any activation function which is infinitely differentiable in any interval can make ELM model fit any object function with probability one under certain conditions. However, there is still no clear guidance on the selection of activation function for different problems.
(4) *Advanced algorithm*. Nowadays, multi-task learning strategy (MTL) has been applied to neural networks and shows the huge development potential. The core idea of MTL is that learning a task together with other related tasks at the same time via a shared representation to improve the performance of model [67]. Specifically, for neural networks, the hidden nodes are shared among multiple tasks while the output nodes are independent for different tasks. [68–71] have shown that integrating MTL into the training process of deep learning can effectively improve the performance of deep neural network. NNRW is also extended to the scenario of MTL and the generalization ability is greatly improved [72–74]. However, what to share and how to design the corresponding network architecture for different

Please cite this article as: W. Cao et al., A review on neural networks with random weights, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.040

problems are needed to be further studied, especially in the deep architecture cases.

3. Deep neural network with random weights and its applications

Deep neural network, aka deep learning, has produced a lot of breakthrough results in recent years [75]. Some notable applications include speech recognition [76,77], image recognition [78,79], customer review sentence sentiment classification [80], and 3D human poses recovery [81–83], etc. One of the reasons for success is that the network structure is getting deeper and deeper. It is well known that feature selection or extraction plays an essentially important role on the learning process. Facts show that the multilayer structure can do better than the shallow network in learning high-level abstractions from complex tasks, such as computer vision, speech recognition, and natural language processing. In other words, Thin + Tall architectures, that is, multilayer architectures, in which each hidden layer do not have a lot of nodes, can be much more efficient than Fat + Short architectures, that is, shallow architectures, in which sometimes only single hidden layer with a lot of nodes [5,84,85]. As mentioned in Section 1, deep learning is often considered as a supervised fine-tuning process with unsupervised initialization. The multilayer architecture is treated as a whole and all the hidden parameters are trained multiple times in a similar way as the BP-based method does. There are a lot of hidden parameters in a typical deep neural network architecture, which means iteratively fine tuning all the hidden parameters will be very time-consuming [3–7].

As discussed in Section 1, NNRW can achieve much faster speed than BP-based methods with acceptable accuracy. Therefore, it is a promising way to incorporate NNRW into the existing deep learning architectures. In the rest of this section, we will show the evolvement of deep neural network with random weights (DNNRW) and its applications, which include Auto-encoder (AE) with random weights, restricted Boltzmann machines (RBN) with random weights, and Convolutional neural networks (CNN) with random weights. Most of the top-level algorithms in deep learning are based on AE, RBM, and CNN. In general, these three algorithms are mainly used for pre-training in deep learning. They can effectively learn transformations from a low-level representation to a high-level one, especially in non-linear cases. Compared with the traditional linear method such as principal component analysis (PCA), they can learn much more significant semantic meanings from raw data. In addition, perhaps more importantly, they can be stacked to form a deep structure, which in turn makes them more powerful. Once deep network is pre-trained, input data will be transformed to a better representation and can be more effectively used for classification.

3.1. Auto-encoder network

Auto-encoder (AE) is a simple 3-layer neural network and mainly used to learn a representation from a dataset by using unsupervised learning. The basic structure of AE is shown in Fig. 5.

In a typical AE, the target values are set to be equal to the inputs, and the number of hidden nodes is much less than number of input nodes. The basic idea is that the hidden neurons are able to extract relevant features from the training data by minimizing the reconstruction error.

Hinton et al. [3] proposed a pre-training method for deep learning based on AE and restricted Boltzmann machine in 2006. This method can effectively improve the training efficiency of deep learning. Their idea is that hidden parameters are initialized by AE from the first hidden layer and the hidden output of previous AE is used as the input of next AE. When unsupervised learning (AE) is completed, all the parameters are fine tuned by using the BP-based method. In this way, they can alleviate the problem of gradient dispersion that often occurs in the training process of deep learning. Subsequently, more and more neural networks based on AE have been developed, such as stacked auto-coder-based (SAE) [5] and stacked denoising auto-coder-based (SDAE) [4].

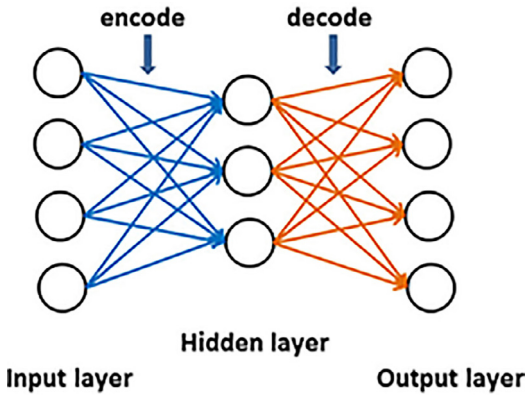

Fig. 5. The structure of Auto-encoder.

Because auto-encoder neural network is an unsupervised learning algorithm with back-propagation, it is inevitable to inherit the shortcomings of the BP-based method, which may affect the performance of deep neural network. Some researchers proposed to take advantage of NNRW to resolve the problem and have made some progress.

Kasun et al. [86] designed a novel auto-encoder based on ELM (ELM-AE), which represents features with singular values. Unlike original ELM, the weights and biases of the hidden nodes are required to be orthogonal, and the input is equal to the output. Based on ELM-AE, they stack it to form a deep architecture (named ML-ELM) by using similar methods like SAE. Unlike the traditional SAE, ML-ELM does not require global fine tuning. Compared with other deep neural networks such as deep belief network [3], ML-ELM can achieve faster speed with higher accuracy on MNIST dataset [87].

Cecotti conducted more verification tests for ML-ELM on four handwritten character databases [88]. The results confirm that ML-ELM has great advantages in accuracy and execution time. As the number of hidden layers increase, the performance of ML-ELM increases until the number of hidden layers reaches a certain number.

Tang et al. [89] optimized ML-ELM with a new ELM-AE model and proposed hierarchical ELM (H-ELM). Compared with ELM-AE proposed in [86], Tang et al. adopted L1-norm optimization instead of L2-norm optimization used in [86]. Because of using the L1 penalty, the new ELM-AE can obtain more sparse and meaningful hidden features. In addition, H-ELM does not require the initial values of hidden parameters to be orthogonal. Several experimental results on car detection, gesture recognition, and real-time object tracking show that H-ELM achieves more robust and better performance when compared with ML-ELM. Iosifidis et al. [90] further applied H-ELM to solve supervised subspace learning problems.

Different from [89], Sun et al. [91] combined the original ELM–AE with manifold regularization and proposed generalized extreme learning machine auto-encoder (GELM–AE). Compared with the original ELM–AE, GELM–AE has stronger ability to extract more relevant features for clustering. It also shows that GELM–AE can be stacked to form a deep structure.

Zhang et al. [92] introduced a local denoising criterion into ELM–AE and proposed ELM denoising auto-encoder (ELM–DAE). They pointed out that the ELM–DAE can extract higher level representations than original ELM–AE. Then the authors stacked

Please cite this article as: W. Cao et al., A review on neural networks with random weights, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.040

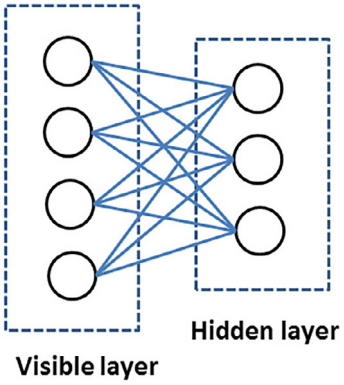

Fig. 6. The structure of RBM.

ELM–DAE to create a deep architecture named Denoising ML-ELM and reported its effectiveness in supervised learning and semi-supervised learning problems.

Hu et al. [93] proposed a stacked deep neural network based on unsupervised extreme learning machines [94] to deal with unsupervised problems. They showed that this architecture can yield a better embedding space for clustering. And their method runs much faster than deep auto-encoder (DA) and stacked auto-encoder (SAE).

In addition, Gu et al. [95] designed a wireless localization system based on the combination of deep neural network embedded with semi-supervised learning and ELM. The former can be used to extract high-level abstract feature from a lot of unlabeled data and ELM is used to make a quick classification.

3.2. Restricted Boltzmann machines

Restricted Boltzmann machine (RBM) is a probabilistic graphical model based on stochastic neural network, which can be used to learn a probability distribution from training data. RBM shares a similar idea with AE, which also can be stacked as building blocks of multi-layer learning architectures, aka deep belief networks (DBNs) [3]. A typical structure of RBM is shown in Fig. 6.

A simple RBM has 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. The visible layer is used to receive the input data and the hidden layer is used to generate a new feature vector. Details of the training process are given in [3]. RBM has been successfully applied to dimensionality reduction, feature learning, and classification, etc.

Hinton et al. [76] showed that the deep neural network with RBM-based weights initialization can achieve better performance in speech recognition than the deep neural network with random weights initialization. One explanation is that RBM-based pre-training can receive a better starting point for parameters tuning to approach to the global optimum. However, here the deep neural network adopted BP-based method for weights tuning. And thus, the learning process is time-consuming.

Rosa et al. [96] merged RBM and randomized algorithms and proposed a deep structure for nonlinear system identification. In this system, the distributions of the hidden weights are trained by using input data and RBM. It shows that combining RBM and the randomized algorithm gives better performances for nonlinear system identification.

In [97], a DBN–ELM structure for image classification was proposed, where DBN is used as features extractor and ELM is trained on DBN-learnt features as the classifier. The DBN + ELM structure can achieve much faster speed and higher accuracy than DBN with other classifiers such as SVM. The similar conclusion can be verified by Han et al. [77].

Zhang et al. [98] proposed a hybrid architecture based on ELM with Manifold Regularization and the semi-restricted Boltzmann machine (SRBM), named IELM–DFE. In IELM–DFE, the SRBM models were stacked to a DBN model for feature extraction, the last hidden layer of this DBN model was used as the hidden layer of ELM, and the output weights were calculated by Manifold Regularization ELM model. The authors showed that IELM–DFE could perform well in classification tasks.

3.3. Convolutional neural networks

Deep neural networks based on convolution have been widely used in computer vision, speech recognition, and natural language processing, etc. Since LeCun et al. [87] proposed a deep architecture, LeNet-5, based on convolutional neural network (CNN) in 1998, many deeper neural networks have been proposed, such as AlexNet [78] and VGG [79]. CNN can efficiently deal with image processing problems due to its unique features, including shared weights, sub-sampling, and local receptive fields. In general, a simplest CNN includes an input layer, a convolutional layer, a pooling layer, a fully-connected layer, and an output layer. The input layer holds the raw pixel values of the image; the convolutional layer computes the output of nodes that are connected to local regions in the input layer; the pooling layer performs the down-sampling operation along the spatial dimensions; the fully-connected layer computes the class scores; and the output layer gives the final results. In this way, we can create a deep architecture by alternately stacking the convolutional layer and pooling layer. A typical CNN architecture is shown in Fig. 7.

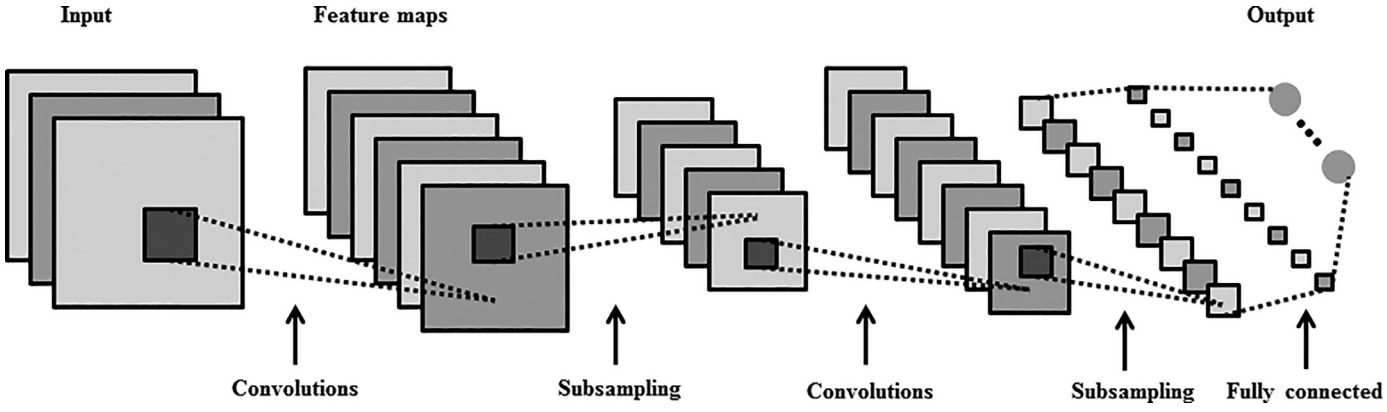

Fig. 7. A typical CNN architecture.

Please cite this article as: W. Cao et al., A review on neural networks with random weights, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.040

Although CNN has been proven to be effective, how to effectively train such deep models is still an unsolved problem.

By conducting thorough experiments, Jarrett et al. [99] found that random filters used in the two-stage feature extraction system can achieve comparative accuracy on the Caltech-101 dataset [100] compared to the approach using unsupervised pre-training and exact fine tuning of the filters. The discovery is of great practical significance, which means that the network architecture plays a more important role than hidden parameters in deep learning. In other words, not all the hidden parameters need to be fine-tuned if we are able to design a proper deep architecture. In this way, the computing efficiency can be greatly improved. A similar discovery can be found in [101].

Zhang et al. [102] combined the RVFL architecture and convolutional neural network and proposed the convolutional random vector functional link (CRVFL) neural network for solving visual tracking problems. In CRVFL, the convolutional filters are randomly initialized and kept same, only the parameters in the fully connected layers need to be learned. Compared with traditional CNNs, this method does not suffering from global fine-tuning and the system is not sensitive to the hyper-parameters such as learning rate, size of mini-batch, and epochs. It was also shown that CRVFL runs on a CPU-based computer can achieve comparable accuracy to the traditional deep learning approaches running on a GPU-based computer, but the learning time is significantly reduced.

Zeng et al. [103] designed a hybrid deep architecture for traffic sign recognition, where CNN acts as a feature extractor and ELM was trained on CNN-learnt features as the classifier. They showed that this architecture has three advantages: low model complexity, high performance, and fast speed. Similar architectures have been applied to handwritten digit recognition [104–106], lane detection task [107], 3D feature learning [108], etc.

3.4. Remarks

Auto-encoder is an efficiency feature extraction technology which can be easily stacked to deep architecture for obtaining high-level representation. In traditional DAE, all the hidden parameters are iteratively tuned according to BP-based method until the predefined stopping criterion is satisfied. This method suffers from several disadvantages, such as low convergence rate, time consuming, high computational complexity, etc. The introduction of NNRW effectively improves the efficiency of this training process. In the hybrid architecture, the hidden parameters are obtained based on non-iterative method. Compared with traditional DAE, the new scheme can achieve much faster training speed and lower computational complexity.

The key idea of the RBN+NNRW deep architecture is that replace the random weight assignment with RBM-based weight initialization and keep the weights of output layer nodes being obtained analytically. It is noted that this method is a non-iterative technique. In this way, the training efficiency can be improved significantly and the computational complexity is reduced markedly.

CNN does well in image feature extraction, while NNRW can achieve fast training speed in classification problems. The hybrid deep architecture of CNN and NNRW can effectively improve the efficiency of BP-based deep CNN. However, there are still several open problems need to be studied. First, the key parameters in CNN are set manually, such as the number of filters, the size of convolutional kernel, etc. Although some researchers have tried to randomly select some of these parameters [99,101], there is no clear criterion to guarantee the performance of deep learning models. Second, in most cases, only the single hidden layer NNRW are used as classifiers in the hybrid deep architecture, whether multiple hidden layer NNRW can be applied in this architecture remains untouched. Third, many applications mentioned above have shown that NNRW are effective for small training datasets, while CNN requires large datasets to exert its powerful feature extraction capabilities. CNN can extract high-level feature representation from raw data, however, what comes with this advantage is more parameters needed to be tuned. And thus large datasets are always required to train a model. Whether the combination of CNN and NNRW can reduce the scale of parameters and make CNN effective on smaller training datasets needs to be studied.

In addition, it is found in [109,110] that the mechanism of random weight assignment can be potentially used to handle reinforcement learning problems. Moreover, other advanced feature selection algorithms [111,112] also have the potential to enhance the performance of NNRW model and are worthy to further study. Similar to the shallow NNRW, the deep NNRW has the advantages of significantly improving the training efficiency, which is an extremely critical problem for gradient based training of large scale feed-forward neural networks with deep architecture. The major disadvantages of deep NNRW are that

(1) During the training process there are not weight-feedbacks, which may play an indispensable role for hierarchical feature extraction in gradient based training of deep networks
(2) The feature extraction for a deep NNRW with layers is completed without feedbacks in the first layers, while the prediction model is built at the layer. Unfortunately, the number of nodes at the last layer is usually very big.

4. Concluding remarks

In this paper, we present a thorough survey on the evolvement of feed-forward neural networks with random weights (NNRW), especially its applications in deep learning. In NNRW, due to the weights and the threshold of hidden layer are randomly selected and the weights of output layer are obtained analytically, NNRW can achieve much faster learning speed than BP-based methods. As described above, NNRW have been widely applied to many applications. Traditional deep learning has produced lots of breakthrough results in recent years. However, it suffers from several notorious problems, such as numerous parameters that need to be tuned, high requirements for computing resources, low convergence rate, and high computational complexity, etc. This paper has shown that the combination of traditional deep learning and NNRW can greatly improve the computing efficiency of deep learning.

However, there are still several open problems need to be addressed, such as how to determine the randomization range and the type of distribution of the hidden weights? It is well known that, the randomization range and the type of distribution of the hidden weights have significant impact on the performance of NNRW.However, there is no clear criterion to guide the selection of the hidden weights. In most cases, the authors directly set the randomization range to an empirical range (i.e., $[-1, 1]$). But this range can not guarantee the optimal performance of NNRW [15]. In addition, NNRW have shown good generalization performance on the problems with higher noise, how to prove it in theory and estimate the oscillation bound of the generalization performance are not clear. Moreover, [38] shows that NNRW are inherently consistent with the mechanism of biological learning, one of the most foundational abilities of biological learning is to handle complex problems with small samples (e.g., for a kid, just a few cat pictures are needed, to learn and grasp the features of the cat), whether NNRW have such a capability have not been fully verified by theories and applications (i.e., whether NNRW can effectively handle the small datasets with high dimensions needs further investigation). In addition to the content mentioned in this article, this type of training mechanism also can be applied

Please cite this article as: W. Cao et al., A review on neural networks with random weights, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.040

to other classical neural networks, such as radial basis function network and kernel-based methods. For more information, you can refer to [17,18]. Zhang and Suganthan [17] give a survey on radial basis function network with random weights and kernel-based methods with random weights. In [18], the authors give a concise review for recurrent neural networks with random weights and randomized kernel approximations.

The following problems may be interesting for further research:

(1) Studying the impact of the randomization range and the type of distribution of the hidden parameters. Parameter selection plays an important role in the performance and stability of NNRW model. And thus, a thorough study of this problem will be of great value.
(2) Improving NNRW algorithms to handle the problems with small samples. Complex problems with small samples are challenging problems for traditional learning algorithms, not for biological learning. It is interesting to study whether NNRW have the similar ability like biological learning.
(3) Giving a rigorously theoretical proof for the effectiveness of random feature mapping in deep NNRW. Random feature mapping plays a key role in NNRW, which ensure the universal approximation capability and the generalization performance of NNRW. It is worth studying the role of random feature mapping in deep NNRW cases.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (61672358, 71371063) and by Basic Research Project of Knowledge Innovation Program in Shenzhen (JCYJ20150324140036825).

References

[1] Y. Chauvin, D.E. Rumelhart, Backpropagation: Theory, Architectures, and Applications, Psychology Press, 1995.
[2] Wikipedia, Alphago Versus Lee Sedol, Wikipedia, (https://en.wikipedia.org/wiki/AlphaGo_versus_Lee_Sedol). Accessed March 25, 2017.
[3] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[4] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.
[5] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al., Greedy layer-wise training of deep networks, Adv. Neural Inf. Process. Syst. 19 (2007) 153.
[6] Y. Bengio, et al., Learning deep architectures for AI, Found. Trends® Mach. Learn. 2 (1) (2009) 1–127.
[7] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[8] 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.
[9] J.-Y. Li, W. Chow, B. Igelnik, Y.-H. Pao, Comments on "stochastic choice of basis functions in adaptive function approximation and the functional-link net"[with reply], IEEE Trans. Neural Netw. 8 (2) (1997) 452–454.
[10] G.-B. Huang, L. Chen, C.K. Siew, et al., Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.
[11] C. Deng, G. Huang, J. Xu, J. Tang, Extreme learning machines: new trends and applications, Sci. China Inf. Sci. 58 (2) (2015) 1–16.
[12] G. Huang, G.-B. Huang, S. Song, K. You, Trends in extreme learning machines: a review, Neural Netw. 61 (2015) 32–48.
[13] Y.-H. Pao, Y. Takefuji, Functional-link net computing: theory, system architecture, and functionalities, Computer 25 (5) (1992) 76–79.
[14] W.F. Schmidt, M.A. Kraaijveld, R.P. Duin, Feedforward neural networks with random weights, in: Proceedings of the 11th IAPR International Conference on Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, IEEE, 1992, pp. 1–4.
[15] L. Zhang, P. Suganthan, A comprehensive evaluation of random vector functional link networks, Inf. Sci. 367 (2016) 1094–1105.
[16] M. Li, D. Wang, Insights into randomized algorithms for neural networks: practical issues and common pitfalls, Inf. Sci. 382 (2017) 170–178.
[17] L. Zhang, P. Suganthan, A survey of randomized algorithms for training neural networks, Inf. Sci. 364 (2016) 146–155.
[18] S. Scardapane, D. Wang, Randomness in neural networks: an overview, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 7 (2) (2017).
[19] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (6) (1958) 386.
[20] Y.-H. Pao, S.M. Phillips, The functional link net and learning optimal control, Neurocomputing 9 (2) (1995) 149–164.
[21] Y.-H. Pao, G.-H. Park, D.J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6 (2) (1994) 163–180.
[22] C.P. Chen, J.Z. Wan, A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction, IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 29 (1) (1999) 62–72.
[23] G.H. Park, Y.H. Pao, Unconstrained word-based approach for off-line script recognition using density-based random-vector functional-link net, Neurocomputing 31 (1) (2000) 45–65.
[24] S. Scardapane, D. Comminiello, M. Scarpiniti, A. Uncini, A semi-supervised random vector functional-link network based on the transductive framework, Inf. Sci. 364 (2016) 156–166.
[25] J.M. Martínez-Villena, A. Rosado-Muñoz, E. Soria-Olivas, Hardware implementation methods in random vector functional-link networks, Appl. Intell. 41 (1) (2014) 184–195.
[26] D. Husmeier, J.G. Taylor, Neural networks for predicting conditional probability densities: improved training scheme combining EM and RVFL, Neural Netw. 11 (1) (1998) 89–116.
[27] D. Husmeier, The bayesian evidence scheme for regularizing probability-density estimating neural networks, Neural Comput. 12 (11) (2000) 2685–2717.
[28] M. Alhamdoosh, D. Wang, Fast decorrelated neural network ensembles with random weights, Inf. Sci. 264 (2014) 104–117.
[29] S. Scardapane, D. Wang, M. Panella, A. Uncini, Distributed learning for random vector functional-link networks, Inf. Sci. 301 (2015) 271–284.
[30] B.-S. Lin, B.-S. Lin, F.-C. Chong, F. Lai, A functional link network with higher order statistics for signal enhancement, IEEE Trans. Signal Process. 54 (12) (2006) 4821–4826.
[31] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, Vol. 2, IEEE, 2004, pp. 985–990.
[32] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (16) (2007) 3056–3062.
[33] G.-B. Huang, An insight into extreme learning machines: random neurons, random features and kernels, Cognit. Comput. 6 (3) (2014) 376–390.
[34] G.-B. Huang, Z. Bai, L.L.C. Kasun, C.M. Vong, Local receptive fields based extreme learning machine, IEEE Comput. Intell. Mag. 10 (2) (2015) 18–29.
[35] 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.
[36] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[37] L.L.C. Kasun, Y. Yang, G.-B. Huang, Z. Zhang, Dimension reduction with extreme learning machine, IEEE Trans. Image Process. 25 (8) (2016) 3906–3918.
[38] G.-B. Huang, What are extreme learning machines? filling the gap between frank rosenblatt's dream and john von neumann's puzzle, Cognit. Comput. 7 (3) (2015) 263–278.
[39] M. Rigotti, O. Barak, M.R. Warden, X.-J. Wang, N.D. Daw, E.K. Miller, S. Fusi, The importance of mixed selectivity in complex cognitive tasks, Nature 497 (7451) (2013) 585–590.
[40] O. Barak, M. Rigotti, S. Fusi, The sparseness of mixed selectivity neurons controls the generalization–discrimination trade-off, J. Neurosci. 33 (9) (2013) 3844–3856.
[41] S. Fusi, E.K. Miller, M. Rigotti, Why neurons mix: high dimensionality for higher cognition, Curr. Opin. Neurobiol. 37 (2016) 66–74.
[42] J. Xie, C. Padoa-Schioppa, Neuronal remapping and circuit persistence in economic decisions, Nat. Neurosci. 19 (6) (2016) 855–861.
[43] E.L. Rich, J.D. Wallis, What stays the same in orbitofrontal cortex, Nat. Neurosci. 19 (6) (2016) 768–770.
[44] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.
[45] Y.-L. He, X.-Z. Wang, J.Z. Huang, Fuzzy nonlinear regression analysis using a random weight network, Inf. Sci. 364 (2016) 222–240.
[46] R. Finker, I. del Campo, J. Echanobe, V. Martínez, An intelligent embedded system for real-time adaptive extreme learning machine, in: Proceedings of the 2014 IEEE Symposium on Intelligent Embedded Systems (IES), IEEE, 2014, pp. 61–69.
[47] L.F. Cambuim, R.M. Macieira, F.M. Neto, E. Barros, T.B. Ludermir, C. Zanchettin, An efficient static gesture recognizer embedded system based on ELM pattern recognition algorithm, J. Syst. Archit. 68 (2016) 1–16.
[48] N.L. Azad, A. Mozaffari, A. Fathi, An optimal learning-based controller derived from hamiltonian function combined with a cellular searching strategy for automotive coldstart emissions, Int. J. Mach. Learn. Cybern. 8 (3) (2017) 955–979.
[49] A.A. Mohammed, R. Minhas, Q.J. Wu, M.A. Sid-Ahmed, Human face recognition based on multidimensional PCA and extreme learning machine, Pattern Recognit. 44 (10) (2011) 2588–2597.
[50] J. Lu, J. Zhao, F. Cao, Extended feed forward neural networks with random weights for face recognition, Neurocomputing 136 (2014) 96–102.
[51] W. Wan, Z. Zhou, J. Zhao, F. Cao, A novel face recognition method: using random weight networks and quasi-singular value decomposition, Neurocomputing 151 (2015) 1180–1186.

Please cite this article as: W. Cao et al., A review on neural networks with random weights, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.040

[52] R. Minhas, A. Baradarani, S. Seifzadeh, Q.J. Wu, Human action recognition using extreme learning machine based on visual vocabularies, Neurocomputing 73 (10) (2010) 1906–1917.
[53] Z. Huang, Y. Yu, J. Gu, H. Liu, An efficient method for traffic sign recognition based on extreme learning machine, IEEE Trans. Cybern. 47 (4) (2017) 920–933.
[54] D.D. Wang, R. Wang, H. Yan, Fast prediction of protein–protein interaction sites based on extreme learning machines, Neurocomputing 128 (2014) 258–266.
[55] G. Wang, Y. Zhao, D. Wang, A protein secondary structure prediction framework based on the extreme learning machine, Neurocomputing 72 (1) (2008) 262–268.
[56] F. Cao, B. Liu, D.S. Park, Image classification based on effective extreme learning machine, Neurocomputing 102 (2013) 90–97.
[57] X. Jiang, J. Liu, Y. Chen, D. Liu, Y. Gu, Z. Chen, Feature adaptive online sequential extreme learning machine for lifelong indoor localization, Neural Comput. Appl. 27 (1) (2016) 215–225.
[58] Q. He, X. Jin, C. Du, F. Zhuang, Z. Shi, Clustering in extreme learning machine feature space, Neurocomputing 128 (2014) 88–95.
[59] S. Ding, N. Zhang, J. Zhang, X. Xu, Z. Shi, Unsupervised extreme learning machine with representational features, Int. J. Mach. Learn. Cybern. 8 (2) (2017) 587–595.
[60] P. Liu, Y. Huang, L. Meng, S. Gong, G. Zhang, Two-stage extreme learning machine for high-dimensional data, Int. J. Mach. Learn. Cybern. 7 (5) (2016) 765–772.
[61] S. Balasundaram, D. Gupta, On optimization based extreme learning machine in primal for regression and classification by functional iterative method, Int. J. Mach. Learn. Cybern. 7 (5) (2016) 707–728.
[62] M.M. Baig, M.M. Awais, E.-S.M. El-Alfy, Adaboost-based artificial neural network learning, Neurocomputing 248 (2017) 120–126.
[63] A. Fu, C. Dong, L. Wang, An experimental study on stability and generalization of extreme learning machines, Int. J. Mach. Learn. Cybern. 6 (1) (2015) 129–135.
[64] Y.-S. Hsu, S.-J. Lin, An emerging hybrid mechanism for information disclosure forecasting, Int. J. Mach. Learn. Cybern. 7 (6) (2016) 943–952.
[65] D. Lowe, D.S. Broomhead, Multi-variable functional interpolation and adaptive networks, Complex Syst. 2 (1988) 321–355.
[66] J. Ji, H. Jiang, B. Zhao, P. Zhai, Crucial data selection based on random weight neural network, in: Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2015, pp. 1017–1022.
[67] R. Caruana, Multitask Learning, Mach. Learn. 28 (1) (1997) 41–75.
[68] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 568–576.
[69] Z. Zhang, P. Luo, C.C. Loy, X. Tang, Facial landmark detection by deep multi--task learning, in: Proceedings of the European Conference on Computer Vision, ECCV (6), 2014, pp. 94–108.
[70] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, J. Kim, Rotating your face using multi-task deep neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 676–684.
[71] S. Ruder, An overview of multi-task learning in deep neural networks, arXiv preprint 1706.05098, (https://arxiv.org/abs/1706.05098). Accessed July 10, 2017.
[72] A. Bueno-Crespo, R.-M. Menchón-Lara, J.-L. Sancho-Gómez, Related tasks selection to multitask learning schemes, in: Proceedings of the International Work-Conference on the Interplay Between Natural and Artificial Computation, Springer, 2015, pp. 213–221.
[73] W. Mao, J. Xu, S. Zhao, M. Tian, Research of multi-task learning based on extreme learning machine, Int. J. Uncertain. Fuzziness Knowl. Based Syst. 21 (supp02) (2013) 75–85.
[74] Y. Jin, J. Li, C. Lang, Q. Ruan, Multi-task clustering ELM for VIS-NIR cross–modal feature learning, Multidimens. Syst. Signal Process. 28 (3) (2017) 905–920.
[75] 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.
[76] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[77] 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.
[78] 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.
[79] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint 1409, (https://arxiv.org/abs/1706.05098). Accessed July 10, 2014.
[80] Z. Guan, L. Chen, W. Zhao, Y. Zheng, S. Tan, D. Cai, Weakly-supervised deep learning for customer review sentiment classification, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI, 2016, pp. 3719–3725.
[81] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670.
[82] C. Hong, J. Yu, D. Tao, M. Wang, Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval, IEEE Trans. Ind. Electron. 62 (6) (2015) 3742–3751.
[83] C. Hong, X. Chen, X. Wang, C. Tang, Hypergraph regularized autoencoder for image-based 3d human pose recovery, Signal Process. 124 (2016) 132–140.
[84] F. Seide, G. Li, D. Yu, Conversational speech transcription using context-dependent deep neural networks., in: Proceedings of the Interspeech, 2011, pp. 437–440.
[85] Y. Bengio, Y. LeCun, et al., Scaling learning algorithms towards ai, Large-scale Kernel Mach. 34 (5) (2007) 1–41.
[86] 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.
[87] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[88] H. Cecotti, Deep random vector functional link network for handwritten character recognition, in: Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 3628–3633.
[89] J. Tang, C. Deng, G.-B. Huang, Extreme learning machine for multilayer perceptron, IEEE Trans. Neural Netw. Learn. Syst. 27 (4) (2016) 809–821.
[90] A. Iosifidis, M. Gabbouj, Supervised subspace learning based on deep randomized networks, in: Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2016, pp. 2584–2588.
[91] K. Sun, J. Zhang, C. Zhang, J. Hu, Generalized extreme learning machine autoencoder and a new deep neural network, Neurocomputing 230 (2017) 374–381.
[92] N. Zhang, S. Ding, Z. Shi, Denoising laplacian multi-layer extreme learning machine, Neurocomputing 171 (2016) 1066–1074.
[93] 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.
[94] G. Huang, S. Song, J.N. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Trans. Cybern. 44 (12) (2014) 2405–2417.
[95] Y. Gu, Y. Chen, J. Liu, X. Jiang, Semi-supervised deep extreme learning machine for wi-fi based localization, Neurocomputing 166 (2015) 282–293.
[96] E. de la Rosa, W. Yu, Randomized algorithms for nonlinear system identification with deep learning modification, Inf. Sci. 364 (2016) 197–212.
[97] B. Ribeiro, N. Lopes, Extreme learning classifier with deep concepts, in: Proceedings of the Iberoamerican Congress on Pattern Recognition, Springer, 2013, pp. 182–189.
[98] J. Zhang, S. Ding, N. Zhang, Z. Shi, Incremental extreme learning machine based on deep feature embedded, Int. J. Mach. Learn. Cybern. 7 (1) (2016) 111–120.
[99] K. Jarrett, K. Kavukcuoglu, Y. LeCun, et al., What is the best multi-stage architecture for object recognition? in: Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 2146–2153.
[100] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories, Comput. Vis. Image Underst. 106 (1) (2007) 59–70.
[101] 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 (ICML-11), 2011, pp. 1089–1096.
[102] L. Zhang, P.N. Suganthan, Visual tracking with convolutional random vector functional link network, IEEE Trans. Cybern. PP (99) (2016) 1–11.
[103] Y. Zeng, X. Xu, Y. Fang, K. Zhao, Traffic sign recognition using extreme learning classifier with deep convolutional features, in: Proceedings of the 2015 International Conference on Intelligence Science and Big Data Engineering (IScIDE 2015), Vol. 9242, Suzhou, China, 2015, pp. 272–280.
[104] M.D. McDonnell, T. Vladusich, Enhanced image classification with a fast-learning shallow convolutional neural network, in: Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), IEEE, 2015, pp. 1–7.
[105] S. Pang, X. Yang, Deep convolutional extreme learning machine and its application in handwritten digit classification, Comput. Intell. Neurosci. 2016 (3) (2016) 1–10.
[106] L. Guo, S. Ding, A hybrid deep learning CNN-ELM model and its application in handwritten numeral recognition, J. Comput. Inf. Syst. 11 (7) (2015) 2673–2680.
[107] J. Kim, J. Kim, G.-J. Jang, M. Lee, Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection, Neural Netw. 87 (2017) 109–121.
[108] Z. Xie, K. Xu, W. Shan, L. Liu, Y. Xiong, H. Huang, Projective feature learning for 3D shapes with multi-view depth images, in: Computer Graphics Forum, Vol. 34, Wiley Online Library, 2015, pp. 1–11.
[109] S. Yasini, M.B.N. Sitani, A. Kirampor, Reinforcement learning and neural networks for multi-agent nonzero-sum games of nonlinear constrained-input systems, Int. J. Mach. Learn. Cybern. 7 (6) (2016) 967–980.
[110] S. Yasini, M.B.N. Sistani, A. Karimpour, N. Sistani, A. Karimpour, Erratum to: reinforcement learning and neural networks for multi-agent nonzero-sum games of nonlinear constrained-input systems, Int. J. Mach. Learn. Cybern. 7 (6) (2016) 981.
[111] E.C. Tsang, Q. Hu, D. Chen, Feature and instance reduction for PNN classifiers based on fuzzy rough sets, Int. J. Mach. Learn. Cybern. 1 (7) (2016) 1–11.
[112] M. Marinaki, Y. Marinakis, A bumble bees mating optimization algorithm for the feature selection problem, Int. J. Mach. Learn. Cybern. 4 (7) (2016) 519–538.

Please cite this article as: W. Cao et al., A review on neural networks with random weights, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.040

Weipeng Cao received his bachelor's degree from Hebei Normal University in June 2010. He is currently a Ph.D. student in College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China. Since 2017 he is serving as a research assistant in the School of Engineering and Computer Science, University of the Pacific, California, USA. His research interests include feature extraction, machine learning, and artificial intelligence.

Xizhao Wang received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1998. He is currently a professor of the College of Computer Science and Software Engineering, Shenzhen University, Guangdong, China. He has more than 180 publications, including four books, seven book chapters, and more than 100 journal papers in the IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Systems, Man, and Cybernetics, IEEE Transactions on Fuzzy Systems, etc. His current research interests include fuzzy measures and integrals, feature extraction, and applications of machine learning.

Zhong Ming received the Ph.D. degree in Computer Science from Sun Yat-sen University, Guangzhou, China. He is currently a professor of the College of Computer Science and Software Engineering, Shenzhen University, Guangdong, China. He has published more than 100 high-quality papers in top conferences and journals such as IEEE Transactions on Computers, ACM Transactions on Embedded Computing Systems, ACM Transactions on Autonomous and Adaptive Systems, IEEE International Conference on Data Mining (ICDM), etc. His current research interests include cloud computing, internet of things, software engineering, and artificial intelligence.

Jinzhu Gao received the Ph.D. degree in Computer Science from the Ohio State University in 2004. She is currently an associate professor of Computer Science at the University of the Pacific. Her main research focus is on big data management, analysis, and visualization for collaborative science. Her work has been published in top journals such as IEEE Transactions on Visualization and Computer Graphics, IEEE Transactions on Computers, and IEEE Computer Graphics and Applications.

Please cite this article as: W. Cao et al., A review on neural networks with random weights, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.040