

Research Article

Protein Sequence Classification with Improved Extreme Learning Machine Algorithms

Jiuwen Cao¹ and Lianglin Xiong^{2,3}

¹ Institute of Information and Control, Hangzhou Dianzi University, Zhejiang 310018, China

² School of Mathematics and Computer Science, Yunnan University of Nationalities, Kunming 650500, China

³ School of Mathematics and Statistics, Yunnan University, Kunming 650091, China

Correspondence should be addressed to Jiuwen Cao; caoj0003@e.ntu.edu.sg

Received 17 December 2013; Revised 15 February 2014; Accepted 16 February 2014; Published 30 March 2014

Academic Editor: Tao Huang

Copyright © 2014 J. Cao and L. Xiong. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Precisely classifying a protein sequence from a large biological protein sequences database plays an important role for developing competitive pharmacological products. Comparing the unseen sequence with all the identified protein sequences and returning the category index with the highest similarity scored protein, conventional methods are usually time-consuming. Therefore, it is urgent and necessary to build an efficient protein sequence classification system. In this paper, we study the performance of protein sequence classification using SLFNs. The recent efficient extreme learning machine (ELM) and its invariants are utilized as the training algorithms. The optimal pruned ELM is first employed for protein sequence classification in this paper. To further enhance the performance, the ensemble based SLFNs structure is constructed where multiple SLFNs with the same number of hidden nodes and the same activation function are used as ensembles. For each ensemble, the same training algorithm is adopted. The final category index is derived using the majority voting method. Two approaches, namely, the basic ELM and the OP-ELM, are adopted for the ensemble based SLFNs. The performance is analyzed and compared with several existing methods using datasets obtained from the Protein Information Resource center. The experimental results show the priority of the proposed algorithms.

1. Introduction

Protein sequences (also known as polypeptides) are organic compounds made of amino acids arranged in a linear chain and folded into a globular form. The amino acids in a polymer chain are joined together by the peptide bonds between the carboxyl and amino groups of adjacent amino acid residues. The sequence of amino acids in a protein is defined by the sequence of a gene, which is encoded in the genetic code. As shown in Figure 1, a gene is any given segment along the deoxyribonucleic acid (DNA) that encodes instructions which allow a cell to produce a specific product. Typically, a protein such as an enzyme initiates one specific action.

Due to the wide applications in clinical proteomics and protein bioinformatics, protein sequence analyses have been comprehensively studied in recent years, such as the work presented by Barve et al. [1], Chen et al. [2], Cong et al. [3], Machado et al. [4], Carregari et al. [5], and Liu et al. [6].

Protein sequence analysis generally helps to characterize protein sequences *in silico* and allows the prediction of protein structures and functions. Recent research has shown that the comparative analysis of the protein sequences is more sensitive than directly comparing DNA. Hence, a number of protein sequence databases have been established in the past decades, such as Protein Information Resource (PIR) (<http://pir.georgetown.edu/>), Protein Data Bank (PDB) (<http://www.pdb.org/pdb/home/home.do>), and Universal Protein Resource (UniProt) (<http://www.uniprot.org/>). Hence, it becomes an important and challenging task to efficiently exploit useful information from the large protein sequence dataset for both computer scientists and biologists. As mentioned by Baldi and Brunak [7], protein sequence classification plays an important role in protein sequence analysis on the account of those protein sequence members consisting of a same protein superfamily are evolutionally related and functionally and structurally relevant to each other. Precisely classifying a member protein sequence

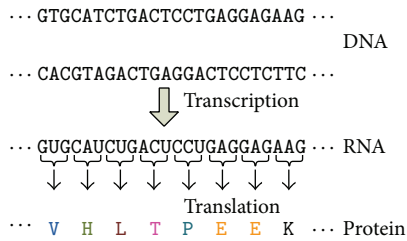


FIGURE 1: The DNA sequence of a gene encodes the amino acid sequence of a protein.

into a superfamily protein would show the benefit that it only needs to carry out some molecular analysis within a particular superfamily instead of the analysis on all the individual member protein sequences. Generally, two protein sequences are classified into the same category if their feature patterns extracted by sequence alignment algorithms show high homology. Lots of alignment algorithms have been proposed in the past few years to identify the class of the unseen protein sequence based on comparing it with some known protein sequences and calculating their similarities, such as iPro-Class (<http://pir.georgetown.edu/>), SAM (SAM: Sequence Alignment and Modeling Software System, Baskin Center for Computer Engineering and Science, <http://www.cse.ucsc.edu/researchcompbio/>), and MEME (MEME: Multiple Expectation Maximization for Motif Elicitation UCSD Computer Science and Engineering, <http://meme.sdsc.edu>). However, it is a very time-costing work to compare the testing protein sequence with all the existing identified protein sequences, especially when the database is large and the length of the unseen protein sequence is long. Therefore, establishing an efficient and intelligent classification system to exactly label the testing protein sequence in a large database becomes urgent and useful.

A number of methods have been developed for general signal classifications based on the statistical theory in the past decades, such as decision trees, statistical techniques, support vector machine (SVM), and neural networks (NN). Yang et al. [8] employed the word segmentation method for feature extraction on the protein sequence and then utilized the SVM for classification. Beside this, Caragea et al. [9] used the hashing function to reduce the dimension of the protein sequence feature vector and then performed classification with SVM. Alternative to using the SVM method, neural networks are another popular method for protein sequences classification in terms of the following two reasons: (i) as the features of protein sequences are generally distributed in a high dimensional space with complex characteristics, it is usually difficult to find a satisfactory model using the statistical or parameterized approaches, and (ii) neural networks are able to process the raw continuous values fed into the model. A lot of research works based on neural networks for protein sequences classification have been done in the last few years, such as Wang et al. [10, 11] and Wang and Huang [12]. Wang et al. [10] proposed a modular radial basis function

(RBF) neural network classifier for the protein sequences with improved classification criteria, and two heuristic rules were presented for the decision-making to enhance the classification reliability. A generalized radial basis function (GRBF) neural network architecture that generates a set of fuzzy classification rules was developed for protein sequences classification by Wang et al. [11]. Most of the previous papers in protein sequence classifications usually chose the gradient based algorithm for neural networks, which are time-consuming in general. Hence, a computationally efficient tuning-free algorithm named extreme learning machine (ELM) for single hidden layer feedforward neural networks (SLFNs), which was recently proposed by Huang et al. [13, 14] and further improved by Huang et al. [15], was applied for protein sequences classification by Wang and Huang [12]. The experimental results given by Wang and Huang [12] have shown that the ELM algorithm learns thousands times faster than the conventional gradient based method (also known as backpropagation (BP), which was developed by Levenberg [16] and Marquardt [17]) with a higher classification rate in protein sequences. To enhance the classification performance and keep the training time in an acceptable level, a self-adaptive evolutionary ELM (SaE-ELM) which utilized the self-adaptive differential evolutionary to update the hidden neuron parameters in the ELM neural network has been presented in Cao et al. [18].

Although the basic ELM and its invariant SaE-ELM have been employed and discussed for protein sequence classification by Wang and Huang [12] and Cao et al. [18], respectively, there are still a lot of rooms for improvements. As presented and discussed by Wang and Huang [12], although ELM learning is much faster than the conventional BP algorithm on the protein sequence dataset, the improvement of classification rate is relatively small. With this objective, we study classification performance of protein sequence based on recent improved ELM algorithms in this paper. The contributions of the paper are threefold. First, the recent robust and generic algorithm named the optimal pruned ELM (OP-ELM) developed by Miche et al. [19] is utilized for protein sequence classification in this paper, where the multiresponse sparse regression (MRSR) technique developed by Simila and Tikka [20] and the leave-one-out (LOO) validation criterion are employed in the OP-ELM for the selection of an appropriate number of hidden neurons. Second, the ensemble based SLFNs network structure is proposed and the majority voting method is adopted to further enhance the protein sequence classification performance. Third, both the basic ELM and the OP-ELM are used as the training algorithms for each ensemble in the new structure. Thus, two algorithms named the voting based ELM (V-ELM) and the voting based OP-ELM (VOP-ELM) are developed for protein sequence classifications. The performance of all the proposed methods is analyzed using the protein sequence database from the Protein Information Resource (PIR) center. Simulations results are compared with recent state-of-art algorithms, such as SVM by Hsu and Lin [21], BP by Haykin [22], Marquardt [17], Levenberg [16], and the original ELM by Huang et al. [13, 14, 23].

Organizations of the rest of the paper are as follows. The feature extraction method for biological protein sequence data used by Wang et al. [10, 11] is reviewed in Section 2. The data description of the protein sequences downloaded from the PIR database is also introduced in this section. In Section 3, the recent ELM and its improved method OP-ELM for protein sequence classification using single hidden layer feedforward neural network are first given. The ensemble based SLFNs structure combined with the majority voting method is then proposed for protein sequence classification. The original ELM and the OP-ELM are used as the learning algorithms for each ensemble. Experimental results and performance comparisons are given in Section 4. Discussions and conclusions are drawn in Section 5.

2. Feature Extraction

As described by Wang et al. [10, 11], a protein sequence is generally made from various combinations of 20 amino acids with notations as $\Sigma = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$. For protein sequence classifications, the n -gram features with a pair of values (v_m, c_m) are extracted as the input signals fed to a classifier, where v_m is the feature m and c_m is the count of this feature in a protein sequence with $m = 1, 2, \dots, 20^n$. For instance, the 2-gram features from the set Σ are all the combinations as $(AA, AC, \dots, AY, CA, CC, \dots, CY, \dots, YA, YC, \dots, YY)$. A feature is the number of occurrences of an amino within a protein sequence. Taking a protein sequence *VAAGTVAGT* as an example, the 2-gram features can be extracted and presented as $\{(VA, 2), (AA, 1), (AG, 2), (GT, 2), (TV, 1)\}$. Another commonly used information for protein sequence feature extraction is the 6-letter exchange group. That is, the 6 combinations of the letters from the whole set Σ are formed as $\mathbf{A} = \{H, R, K\}$, $\mathbf{B} = \{D, E, N, Q\}$, $\mathbf{C} = \{C\}$, $\mathbf{D} = \{S, T, P, A, G\}$, $\mathbf{E} = \{M, I, L, V\}$, and $\mathbf{F} = \{F, Y, W\}$. Therefore, using the 6-letter group, the above protein sequence *VAAGTVAGT* can be transformed as *EDDDDED* and its 2-gram features are $\{(DE, 1), (ED, 2), (DD, 5)\}$. Similar to the feature extraction done by Wang et al. [10, 11] and Wang and Huang [12], we use \mathbf{e}_n and \mathbf{a}_n to represent n -gram features from the 6-letter group and the 20-letter set, respectively. Each set of n -gram features from a protein sequence, that is, \mathbf{e}_n and \mathbf{a}_n , is scaled separately to avoid skew in the counts value using the formula: $\bar{x} = (x/(I-n+1))$, where x is the count of the generic gram feature, \bar{x} is the normalized x , which is the inputs of the classifiers, I is the length of the protein sequence, and n is the size of the n -gram features.

A 56-dimensional feature vector, extracted from a protein sequence and comprised from n -gram features of the 6-letter group represented as \mathbf{e}_2 and the 20 letters set represented as \mathbf{a}_1 , is used as the input information of the classifiers. Two protein sequence datasets with ten-super families (classes) were obtained from the PIR databases and denoted as PIR1 and PIR2, respectively. There are 949 protein sequences samples in PIR1 and 534 protein sequences samples in PIR2. The details of the ten superfamilies to be classified are Cytochrome *c* (113/17), Cytochrome *c6* (45/14), Cytochrome

b (73/100), Cytochrome *b5* (11/14), Triosephosphate isomerase (14/44), Plastocyanin (42/56), Photosystem II D2 protein (30/45), Ferredoxin (65/33), Globin (548/204), and Cytochrome *b6-f* complex 4.2K (8/6), where the first digit in the bracket denotes the number of the protein sequences in the PIR1 database and the second digit represents the number of the protein sequences in the PIR2 database, respectively.

3. Methodologies

3.1. SLFN for Protein Sequence Classification

3.1.1. Model Description of SLFN. For the supervised learning in SLFNs, a dataset with input signal features and their associated class category is generally available to train the network parameters. Assuming that the available dataset is $\mathcal{A} = \{(\mathbf{x}_i, t_i)\}_{i=1}^N$, where \mathbf{x}_i , t_i , and N represent the feature vector of the i th protein sequence, its corresponding category index, and the number of protein sequences, respectively; a single hidden layer feedforward neural network (SLFN) with J nodes in the hidden layer can be expressed as

$$\mathbf{o}_i = \sum_{j=1}^J \mathbf{w}_j g(\mathbf{a}_j, b_j, \mathbf{x}_i), \quad i = 1, 2, \dots, N, \quad (1)$$

where \mathbf{o}_i is the output obtained by the SLFN associated with the i th input protein sequence, and $\mathbf{a}_j \in \mathcal{R}^d$ and $b_j \in \mathcal{R}$ ($j = 1, 2, \dots, J$) are parameters of the j th hidden node, respectively. The variable $\mathbf{w}_j \in \mathcal{R}^m$ is the link connecting the j th hidden node with the output layer and $g(\cdot)$ is the hidden node activation function. With all training samples, (1) can be expressed in the compact form as

$$\mathcal{O} = \mathcal{H}\mathcal{W}, \quad (2)$$

where $\mathcal{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_J)$ and \mathcal{O} are the output weight matrix and the network outputs, respectively. The variable \mathcal{H} denotes the hidden layer output matrix with the entry $\mathcal{H}_{ij} = g(\mathbf{a}_j, b_j, \mathbf{x}_i)$. To perform multiclass classification, SLFNs generally utilize the One-Against-All (OAA) method to transform the classification application to a multioutput model regression problem. That is, for a \mathcal{C} -categories classification application, the output label t_i of the protein sequence feature vector \mathbf{x}_i is encoded to a \mathcal{C} -dimensional vector $\mathbf{t}_i = (t_{i1}, t_{i2}, \dots, t_{i\mathcal{C}})^T$ with $t_{ic} \in \{1, -1\}$ ($c = 1, 2, \dots, \mathcal{C}$). If the category index of the protein sequence \mathbf{x}_i is \mathbf{c} , then t_{ic} is set to be 1 while the rest entries in \mathbf{t}_i are set to be -1. Hence, the objective of training phase for the SLFN in (1) becomes finding the best network parameters set $S = \{(\mathbf{a}_j, b_j, \mathbf{w}_j)\}_{j=1, \dots, J}$ such that the following error cost function is minimized:

$$\min_S E = \min_S \|\mathcal{O} - \mathcal{T}\|, \quad (3)$$

where $\mathcal{T} = (\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_N)$ is the target output matrix. The most popular algorithm is the gradient descent based method where the network back-forward errors are used to iteratively update the network parameters. However, the slow learning

Given:

- Training dataset \mathcal{A} ; number of hidden nodes \mathbf{J} , activation function $g(\cdot)$
- (1) Randomly generate \mathbf{a}_j and b_j for $j = 1, \dots, \mathbf{J}$;
 - (2) Calculate \mathcal{H} ;
 - (3) Find \mathcal{W} according to (4).

ALGORITHM 1: ELM [14, 23].

speed and poor learning scalability limit their applications in large datasets and high dimensional signals, such as the protein sequence classification.

3.1.2. ELM. Alternative to iteratively tuning the network parameters, extreme learning machine (ELM), which was recently developed by Huang et al. [13, 14, 23], claims that random hidden node parameters can be utilized for SLFNs and the hidden node parameters may not need to be tuned. It was stated in Huang et al. [14] that a standard SLFN with \mathbf{N} hidden nodes using random input weights and biases and the infinitely differentiable activation function can exactly learn \mathbf{N} arbitrary distinct samples. In such case, the system (2) becomes a linear model and the network parameter matrix can be analytically solved by using the least-square method. That is,

$$\mathcal{W} = \mathcal{H}^\dagger \mathcal{I}, \quad (4)$$

where \mathcal{H}^\dagger is the Moore-Penrose generalized inverse of the hidden layer output matrix \mathcal{H} given by Serre [24]. The universal approximation property of the ELM theory by using random input weights and biases is also presented in Huang et al. [14] to support the algorithm.

In the following, a brief summary of the ELM algorithm is given in Algorithm 1.

As illustrated through simulations given by Huang et al. [14, 23, 25], Liang et al. [26], and Zhang et al. [27], utilizing random hidden parameters in the training phase, ELM not only learns thousand times faster than the BP algorithm and its variants but also has a comparable generalization performance as the conventional gradient descent based methods and SVM. The protein sequence classification based on ELM has been first studied by Wang and Huang [12]. The experimental results obtained by Wang and Huang [12] have shown that ELM performs slight better than the BP algorithm where the improvement of the successful testing classification rate obtained by ELM is around 1%. But ELM runs thousand times faster than BP.

3.1.3. OP-ELM. However, using random hidden node parameters and the tuning-free learning framework in ELM may also bring some issues. For example, the parameters may not be the optimal one and redundant nodes may exist. To address these issues and enhance the performance, many invariants of ELM have been developed in the past several years, such as considering the distribution of input dataset

done by Cao et al. [28] and utilizing the differential evolutionary method for parameter optimization by Zhu et al. [29] and Cao et al. [30]. One of the recent representative improvement methods named the optimal pruned ELM (OP-ELM) is developed by Miche et al. [19] to prune the related neurons with irrelevant variables. To get rid of unuseful hidden nodes of ELM, OP-ELM combines the multiresponse sparse regression (MRSR) by Simila and Tikka [20] with the leave-one-out (LOO) validation method to rank hidden neurons and to select the actual best number of neurons for the model.

The MRSR technique is described as follows. Suppose $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_m] \in \mathcal{R}^{n \times m}$ is the regressor and MRSR adds each column of the regressor matrix one by one to the model $\mathbf{Y}^l = \mathbf{X}\boldsymbol{\beta}^l$, where $\mathbf{Y}^l = [\mathbf{y}_1^l \dots \mathbf{y}_m^l]$ is the target approximation of the model. At the l th step, the weight matrix $\boldsymbol{\beta}^l$ has l nonzero rows. A new nonzero row and a new column of the regressor matrix are added to the model when increasing the steps. With the MRSR method, the hidden neurons \mathbf{h}_i which are the rows of the hidden layer output matrix \mathcal{H} are ranked. For the details of MRSR, one can refer to the work presented by Simila and Tikka [20].

To select the best number of neurons, the LOO validation method is introduced in OP-ELM. Employing the PREdiction Sum of Squares (PRESS) statistics (presented by Bontempi et al. [31] and Myers [32]), the LOO validation error $\varepsilon^{\text{PRESS}}$ is calculated as

$$\varepsilon^{\text{PRESS}} = \frac{t_i - \mathbf{h}_i \boldsymbol{\beta}}{1 - \mathbf{h}_i \mathbf{P} \mathbf{h}_i^T}, \quad (5)$$

where $\mathbf{P} = (\mathcal{H}^T \mathcal{H})^{-1}$. Then, the appropriate number of neurons for the model is taken by evaluating the LOO error versus the used number of neurons. The OP-ELM can be summarized in three steps as found in Algorithm 2.

3.2. Ensemble Based SLFNs for Protein Sequence Classification.

For both ELM and OP-ELM, we can find that the hidden node parameters are randomly assigned once and never updated. With a single SLFN training by ELM or OP-ELM on the protein sequence dataset, the misclassification number of samples may be high. To address this issue, we propose an ensemble based structure of SLFNs for protein sequence classification. Since the original ELM enjoys a much fast training speed for SLFN, it is feasible to employ multiple independent SLFNs to predict the category index with the majority voting method. Rather than relying on single realization of random parameters in ELM, employing the ensemble method would

- (1) Utilize ELM to train the SLFN;
- (2) Rank the hidden neurons of ELM by the MRSR method;
- (3) Decide the proper number of neurons using the LOO validation error in terms of using (5).

ALGORITHM 2: OP-ELM [19].

Given:

- Protein sequence dataset \mathcal{A} ; number of ensembles \mathbf{K}
- (1) Randomly generate \mathbf{K} sets of hidden node parameters, train each SLFN via ELM and obtain the corresponding output weight matrix;
 - (2) For each testing protein sequence \mathbf{x}^{test} , obtain the estimated category indexes for all \mathbf{K} SLFNs constructed in *Step 1*, update (6);
 - (3) Decide the class label of \mathbf{x}^{test} via (7).

ALGORITHM 3: V-ELM.

be able to reduce the misclassification number. The proposed structure of the ensemble based SLFNs for protein sequence classification is given in Figure 2.

As shown in Figure 2, instead of using a single SLFN, multiple independent SLFNs where each one has the same structure and the same hidden node activation function are used. The protein sequence feature vectors are fed to train each SLFN separately. Then, the final category index of the testing sample is decided by majority voting among all the results obtained by these SLFNs. It is apparent that the gradient descent based methods are not suitable to adopt here as the training algorithm for each SLFN. In general, the training time cost by ensemble based SLFNs is linearly increased with a proportion to the number of independent ensembles. Since the conventional gradient descent methods normally suffer from a long training time for the high dimensional feature vectors and large sample sizes, the training time increases dramatically when employing it for multiple ensembles. Therefore, in this paper, we propose to use the basic ELM and the recent OP-ELM as the training algorithm for each SLFN for protein sequence classification as follows.

3.2.1. The Proposed Ensemble Based ELM. The voting based ELM (V-ELM) developed by Cao et al. [33] is proposed for protein sequence classification in this section. The details are described as follows. We assume that \mathbf{K} SLFNs are employed as ensembles. In the first stage, \mathbf{K} sets of hidden parameters $\{(\mathbf{a}_j^k, \mathbf{b}_j^k)_{j=1, \dots, J}\}_{k=1, \dots, \mathbf{K}}$ are randomly generated and the corresponding \mathbf{K} sets of output weight matrix $\{\mathcal{W}^k\}_{k=1, \dots, \mathbf{K}}$ are obtained with (4) using the protein sequence training dataset \mathcal{A} . In the second stage, for each trained ELM denoted as $S^k = \{(\mathbf{a}_j^k, \mathbf{b}_j^k)_{j=1, \dots, J}, \mathcal{W}^k\}_{k=1, \dots, \mathbf{K}}$, the estimated category index of the testing protein sequence sample \mathbf{x}^{test} is obtained. For all \mathbf{K} ensembles, a \mathcal{C} -dimensional vector $\mathcal{L}_{\mathbf{K}, \mathbf{x}^{\text{test}}}$ is used to store all the results where if the category index derived by the k th

($k \in [1, \dots, \mathbf{K}]$) ELM is ι , the value of the corresponding entry ι in the vector $\mathcal{L}_{\mathbf{K}, \mathbf{x}^{\text{test}}}$ is increased by one; that is,

$$\mathcal{L}_{\mathbf{K}, \mathbf{x}^{\text{test}}}(\iota) = \mathcal{L}_{\mathbf{K}, \mathbf{x}^{\text{test}}}(\iota) + 1. \quad (6)$$

In the third stage, the final category index of \mathbf{x}^{test} is determined via

$$\mathbf{c}^{\text{test}} = \arg \max_{\iota \in [1, \dots, \mathcal{C}]} \{\mathcal{L}_{\mathbf{K}, \mathbf{x}^{\text{test}}}(\iota)\}. \quad (7)$$

A proposition is also given by Cao et al. [33] to illustrate the superiority of the V-ELM over the original ELM as follows.

Proposition 1 (see Cao et al. [33]). *Given a standard SLFN with \mathbf{J} hidden nodes, an output function $g(\cdot)$ and a set of training samples $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^N$ assume that the probability of correctly predicting the testing sample \mathbf{x}^{test} using ELM under all different possible hidden node parameters \mathbf{a} and \mathbf{b} is $\mathcal{P}_{ELM}(\mathbf{c} | \mathbf{x}^{\text{test}})$. If the following inequality holds*

$$\mathcal{P}_{ELM}(\mathbf{c} | \mathbf{x}^{\text{test}}) > \max_{\iota \in [1, \dots, \mathcal{C}], \iota \neq \mathbf{c}} \{\mathcal{P}_{ELM}(\iota | \mathbf{x}^{\text{test}})\} \quad (8)$$

where $\mathcal{P}_{ELM}(\iota | \mathbf{x}^{\text{test}})$ is the probability that ELM classifies \mathbf{x}^{test} to category ι that is different from the class \mathbf{c} , then, with a sufficiently large independent training number \mathbf{K} , V-ELM is able to correctly classify \mathbf{x}^{test} with probability one.

A brief summary of the proposed V-ELM for protein sequence classification is described in Algorithm 3.

3.2.2. The Proposed Ensemble Based OP-ELM. Besides using ELM as the training algorithm for each ensemble, we propose the voting based OP-ELM (VOP-ELM) for the protein sequence classification in this section. For each ensemble, the OP-ELM algorithm is incorporated as the training algorithm instead of using the original ELM. Similar to V-ELM, each SLFN ensemble is trained using the OP-ELM method on the

Given:
 Protein sequence dataset \mathcal{A} ; number of ensembles \mathbf{K}
 (1) Utilize OP-ELM to train each SLFN ensemble independently;
 (2) For each testing protein sequence \mathbf{x}^{test} , obtain the estimated category indexes for all \mathbf{K} SLFNs in *Step 1*, update (9);
 (3) Decide the final class label of \mathbf{x}^{test} using (10).

ALGORITHM 4: VOP-ELM.

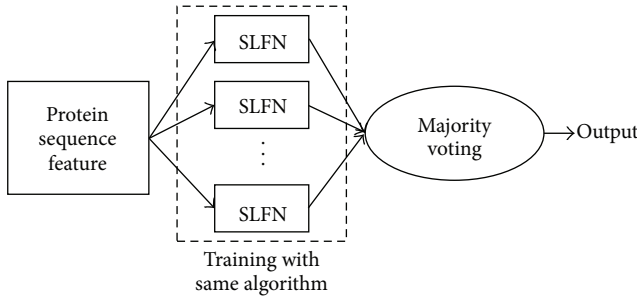


FIGURE 2: Ensemble structure of SLFNs for protein sequence classification.

protein sequence dataset \mathcal{A} independently. After training, the category index of each testing sample \mathbf{x}^{test} by all the \mathbf{K} ensembles is obtained. Similar to (6), the \mathcal{C} -dimensional vector $\mathcal{L}_{\mathbf{K}, \mathbf{x}^{\text{test}}}^{\text{OP}}$ is updated via the following equation:

$$\mathcal{L}_{\mathbf{K}, \mathbf{x}^{\text{test}}}^{\text{OP}}(t) = \mathcal{L}_{\mathbf{K}, \mathbf{x}^{\text{test}}}^{\text{OP}}(t) + 1. \quad (9)$$

Then, the final category index of the testing protein sequence is obtained by

$$\mathbf{c}^{\text{test}} = \arg \max_{t \in [1, \dots, \mathcal{C}]} \{ \mathcal{L}_{\mathbf{K}, \mathbf{x}^{\text{test}}}^{\text{OP}}(t) \}. \quad (10)$$

A brief summary of the proposed VOP-ELM for protein sequence classification is illustrated in Algorithm 4.

We can find that the proposed VOP-ELM would have a better performance than the V-ELM for each ensemble. Compared with V-ELM, the hidden neurons of each ensemble in VOP-ELM are optimized using the OP-ELM given in Section 3.1. However, the drawback of VOP-ELM is that the cost training time may increase as for each SLFN in VOP-ELM, the MRSR technique, and the LOO validation are performed for hidden neurons selection.

4. Experimental Results

In this section, the classification results and discussions on the protein sequence dataset from the Protein Information Resource center are presented. We study the performance in two scenarios. In the first scenario, the PIR1 dataset with 949 samples is fixed as the training dataset while the PIR2 dataset with 534 samples is used as the testing dataset. In the second scenario, the PIR1 and PIR2 datasets are first mixed together and then the training and testing datasets with

949 and 534 samples, respectively, are randomly generated from the mixed dataset. For each protein sequence, a 56-dimensional feature vector is extracted as introduced in Section 2 and hence, the number of nodes in the input layer of the SLFN is 56. The proposed ensemble based VOP-ELM and V-ELM are implemented for the protein sequence classifications. The OP-ELM developed by Miche et al. [19] is also used for comparisons. To compare the classification performance, the original ELM by Huang et al. [14, 23], one of the fastest algorithms of the BP's variants, namely, the Levenberg-Marquardt method by Haykin [22], Levenberg [16], and the SVM by Hsu and Lin [21], is employed for the protein sequence classifications. Three different kernel functions are used as the activation function in the SLFN and SVM, which are the linear kernel function (denoted as \mathbf{L}), the sigmoid kernel function (denoted as \mathbf{S}), and the Gaussian kernel function (denoted as \mathbf{G}), respectively. To maintain the training time under an acceptable level, only $\mathbf{K} = 3$ ensembles of SLFNs are used for VOP-ELM. However, for V-ELM, $\mathbf{K} = 7$ ensembles of SLFNs are employed due to that the V-ELM utilizes the original ELM as training algorithm and it learns much faster than OP-ELM. For the fairness of comparison, all these simulations are executed in the MATLAB 7.4 environment running on an ordinary PC with 3.0 GHz CPU and 4.G RAM memory. Simulations with SVM are carried out using the compiled C-code SVM package: Libsvm (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>) running in the same PC. For SVM, the cost parameter c and the kernel parameter γ are searched in a grid as proposed by Hsu and Lin [21] and the best combination of these parameters is then obtained in terms of the generalized performance. For all these methods, multiple independent trials of simulation are tested and the average results are reported in the paper. For the BP method, 10 trials are used due to its long training time while, for the rest approaches, 50 trials are utilized. The number of hidden nodes for the BP method is 80. When further increasing the number of the nodes, it usually runs out of memory in our PC, which means the computational complexity of the BP algorithm is very high when processing the protein sequence. All the features of the protein sequence are normalized within the region $[-1, 1]$.

4.1. Performance on Fixed Training and Testing Datasets.

In this section, we fix the protein sequence PIR1 as the training dataset while the protein sequence PIR2 is used as the testing dataset. Table 1 shows the comparisons of the average successful testing classification rate (Rate) and its standard

TABLE 1: Successful testing classification rate and standard derivation (rate (%) \pm Dev (%)) comparisons of different classifiers: fixed protein sequence training dataset (pir1) and testing dataset (pir2), where L, S, and G stand for the linear kernel, the sigmoid kernel, and the Gaussian kernel, respectively.

Methods	L	S	G
VOP-ELM	92.51 \pm 0	92.64 \pm 1.6	93.69 \pm 0.89
V-ELM	87.45 \pm 0.01	90.14 \pm 0.71	90.46 \pm 0.69
OP-ELM	91.39 \pm 0.01	91.51 \pm 1.0	92.28 \pm 0.70
SVM	90.63 \pm 0.01	90.63 \pm 0.01	90.63 \pm 0.01
BP	85.77 \pm 0.42	88.39 \pm 1.69	88.77 \pm 1.35
ELM	85.39 \pm 0	87.66 \pm 1.2	87.80 \pm 1.4

derivation (Dev) among multiple trials for all the classifiers. Table 2 gives the corresponding training time cost by all the classifiers and their comparisons.

As highlighted in the boldface in Table 1, the proposed ensemble based VOP-ELM has the highest successful classification rate among all 6 approaches. It can be seen that the proposed VOP-ELM offers an improvement of 7.12% and 1.12%, an improvement of 4.98% and 1.13%, an improvement of 5.89% and 1.41% over the original ELM and the OP-ELM by using the linear kernel L, the sigmoid kernel S, and the Gaussian kernel G, respectively. The V-ELM method performs better than the original ELM and BP in general. The improvement of the classification rate obtained by V-ELM over the original ELM is 2.06%, 2.48%, and 2.66% with the linear kernel, the sigmoid kernel, and the Gaussian kernel, respectively. But the performance of V-ELM is slightly worse than VOP-ELM, OP-ELM, and SVM. It is also worth pointing out that, for all these approaches, using nonlinear kernels (the S and G functions) generally achieves a higher recognition rate than using the linear kernel. As expected, the original ELM has the fastest training speed. As shown in Table 2, the training phase of the original ELM can be finished less than 1 second (s). It learns ten thousand times faster than BP, thousand times faster than SVM and VOP-ELM (the S and G kernels), and hundred times faster than OP-ELM (the S and G kernels). In addition, the training time cost by V-ELM is linearly proportional to the number of ensembles we have used as compared to the training time by the original ELM. With only 7 ensembles, the training phase of V-ELM still can be finished within 1 second, as underlined in Table 1.

In general, a large number of SLFNs used as ensembles usually guarantees a higher classification rate than using a small number of SLFNs as ensembles, but the training time also linearly increases. To illustrate this, three different numbers of ensembles with $K = 5$, $K = 7$, and $K = 15$ are tested on different hidden nodes in each SLFN. The performance is compared with the original ELM. Figures 3 and 4 depict the classification rate and training time of ELM and V-ELM on using the protein sequence PIR1 as training dataset and the PIR2 as the testing dataset. As demonstrated in these two figures, the classification rate increases along with the number of ensembles while the training time also increases gradually. Even though, we can still verify from Figure 4 that V-ELM can finish the training phase within

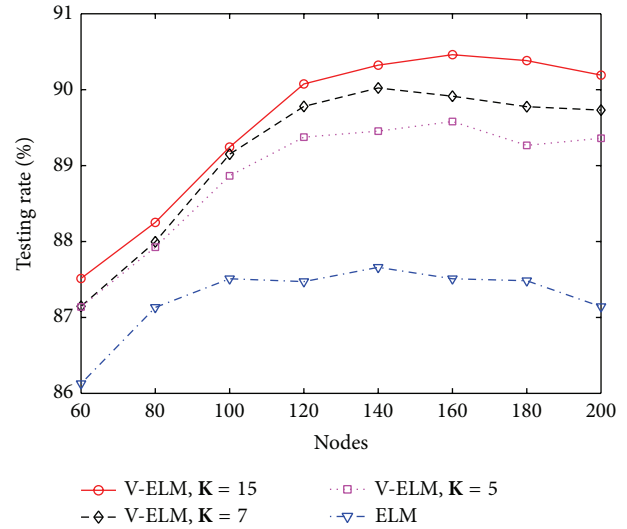


FIGURE 3: The classification rates of ELM and V-ELM w.s.t. different nodes on fixed training and testing protein sequence datasets.

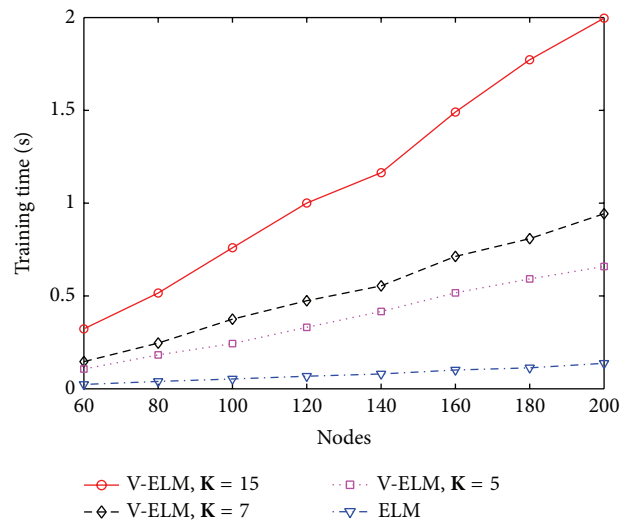


FIGURE 4: The training time of ELM and V-ELM w.s.t. different nodes on fixed training and testing protein sequence datasets.

several seconds in general, which makes it acceptable for large sample sizes protein sequence applications. The performance of VOP-ELM with respect to (w.s.t) different numbers of ensembles is not shown here. This is because when further increasing the numbers of ensembles in VOP-ELM to 7 or more, the training time jumps to hundreds or thousands seconds. It will affect its feasibility of applications on large protein datasets.

4.2. Performance on Randomly Generated Training and Testing Datasets. In this experiment, the protein sequence datasets PIR1 and PIR2 are first mixed into one file after the feature extraction. Then, for each trial, 939 samples are randomly generated from the whole dataset as the training dataset and the rest samples are assigned to the testing dataset. Table 3

TABLE 2: Training time comparisons of different classifiers: fixed protein sequence training dataset (pir1) and testing dataset (pir2), where L, S, and G stand for the linear kernel, the sigmoid kernel, and the Gaussian kernel, respectively.

Methods	L		S		G	
	Training time (s)	Speedup	Training time (s)	Speedup	Training time (s)	Speedup
VOP-ELM	5.5685	105.4	86.80	13.37	95.22	13.93
V-ELM	<i>0.4590</i>	1279.3	<i>0.9020</i>	1286.14	<i>0.8680</i>	1528.34
OP-ELM	1.8588	315.90	30.11	38.53	30.45	43.57
SVM	236.16	2.486	236.67	4.90	243.10	5.46
BP	587.20	1	1160.1	1	1326.6	1
ELM	0.0612	9594.8	0.0799	14519	0.0792	16750

TABLE 3: Successful testing classification rate and standard derivation (rate (%) \pm Dev (%)) comparisons of different classifiers: randomly generated training and testing datasets from the mixed protein sequences, where L, S, and G stand for the linear kernel, the sigmoid kernel, and the Gaussian kernel, respectively.

Methods	L	S	G
VOP-ELM	97.30 \pm 0.73	98.19 \pm 0.70	98.68 \pm 0.71
V-ELM	96.91 \pm 0.71	97.75 \pm 0.64	97.74 \pm 0.59
OP-ELM	95.95 \pm 0.29	96.42 \pm 0.45	97.55 \pm 0.55
SVM	97.17 \pm 0.49	97.28 \pm 0.63	97.33 \pm 0.66
BP	94.93 \pm 0.95	96.29 \pm 0.85	95.54 \pm 0.91
ELM	94.94 \pm 0.75	96.72 \pm 0.85	96.65 \pm 0.63

lists the average successful testing classification rate and the standard derivation of the 6 classifiers among all trials. The training time for the 6 classifiers and their comparisons is shown in Table 4.

From Table 3, it can be seen that the proposed VOP-ELM wins the highest classification rate among all classifiers for all 3 kernels. The VOP-ELM achieves the classification rates of 97.30%, 98.19%, and 98.68% by using the linear, sigmoid, and Gaussian kernels, respectively. The improvements offered by the VOP-ELM method over the original ELM, BP, and OP-ELM are around 2.36%, 2.37%, and 1.35% with the linear kernel, 1.47%, 1.90%, and 1.77% with the sigmoid kernel, and 2.03%, 3.14%, and 1.13% with the Gaussian kernel, respectively. The ensemble based V-ELM performs slightly worse than the VOP-ELM. However, V-ELM is better than the rest methods in general when using the nonlinear kernels. Similar to the experiment in Section 4.1, for the randomly generated training and testing protein sequences, the classifier employing the nonlinear kernel functions (S and G) generally outperforms using the linear kernel function (L).

As shown in boldface in Table 4, ELM is the fastest method and the training phase of the protein sequence dataset takes less than 1 second. However, the conventional BP method takes more than 1 thousand seconds when employing the linear and Gaussian kernels and more than 950 seconds when using the sigmoid kernel. The training time cost by SVM is more than 270 seconds for all the three kernels. Although the original ELM algorithm is employed by the proposed VOP-ELM and OP-ELM, the cost training times jump to dozens of seconds due to utilizing the framework of

searching the optimal hidden neurons and multiple ensembles. With only using 7 ensembles in V-ELM, its training time is at the comparable level as the original ELM (as underlined in Table 4). The enhancement of classification rates obtained by V-ELM over the original ELM is 1.94%, 1.03%, and 1.09% for the linear, sigmoid, and Gaussian kernels, respectively.

To further study the performance of V-ELM w.s.t. the number of ensembles, three different numbers of ensembles $K = 5$, $K = 7$, and $K = 15$ are tested on the randomly generated protein sequence datasets. The classification rates and their corresponding training time are compared with the original ELM by using the sigmoid kernel and the results are depicted in Figure 5 and Figure 6, respectively. As illustrated in these two figures, the classification rates obtained by V-ELM gradually increase when the used ensembles are increased from 5 to 15. All the results obtained by V-ELM are better than the original ELM. Although the training times also increase linearly along with the numbers of ensembles, the training phase for V-ELM still can be finished less than 2 seconds, as shown in Figure 6. Similar to Section 4.1, the performance of the proposed VOP-ELM w.s.t. the number of ensembles is not studied in this section because when more than 7 ensembles are used, the training time cost by VOP-ELM will jump to hundreds or thousands seconds. For the large protein sequence dataset, it may not be feasible.

From Tables 1 and 3, it is interesting to see that although the number of samples used from training dataset is the same, the classification rate obtained by using randomly generated protein sequence training dataset is higher than the one by fixing the PIR1 as the training dataset. The reason behind this may be explained as follows. When fixing the PIR1 as the training dataset, some families of the protein sequences become an imbalance problem, such as the families of Cytochrome c, Cytochrome c6, Triosephosphate isomerase, and Globin as shown in Section 2. In such case, the testing samples have a high risk to be misclassified. However, when generating the training and testing dataset from the whole pool, the samples have equal probability to be partitioned into the training dataset and the testing dataset. Hence, the classification model would be trained with a balance dataset, which may result in a high testing rate.

4.3. Performance on Different Numbers of Ensembles. In this section, we show the relationship between the number of ensembles used in the proposed algorithms and the

TABLE 4: Training time comparisons of different classifiers: randomly generated training and testing datasets from the mixed protein sequences, where L, S, and G stand for the linear kernel, the sigmoid kernel, and the Gaussian kernel, respectively.

Methods	L		S		G	
	Training time (s)	Speedup	Training time (s)	Speedup	Training time (s)	Speedup
VOP-ELM	5.2799	266.44	83.55	11.38	92.00	12.43
V-ELM	0.6326	2223.83	0.6917	1374.51	0.9282	1232.17
OP-ELM	1.8627	755.24	30.93	30.74	32.59	35.09
SVM	278.35	5.05	284.30	3.34	272.41	4.2
BP	1406.8	1	950.75	1	1143.70	1
ELM	0.0746	18858	0.0861	11042	0.0983	11635

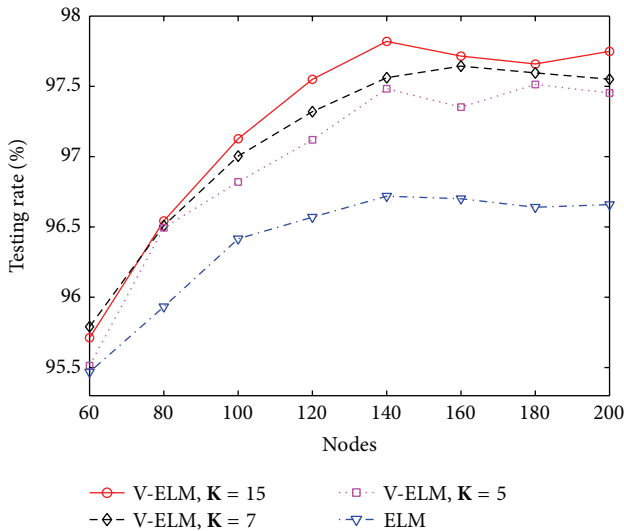


FIGURE 5: The testing rates of ELM and V-ELM w.s.t. different nodes on randomly generated training and testing protein sequence datasets.

classification performance on the protein sequences. The proper number of ensembles which should be adopted in the proposed algorithms is also discussed in the section through the simulations. In the experiments, the number of ensembles is increased from 1 to 51 with an interval 2. The sigmoid kernel is utilized for the experiments as the performance is similar to the one using the other two kernels. The average testing rate for each number of ensembles with 50 independent trials is recorded for both the V-ELM algorithm and the VOP-ELM method. The experiments are running using the PIR1 as the training dataset only. For randomly generated protein sequence dataset, we can find the similar trend of the testing rate and training time on different numbers of ensembles as the one using fixed training dataset. Therefore, the performance on randomly generated training dataset is not repeated here.

Figure 7 depicts the testing rate on different numbers of ensembles for the fixed protein sequence training dataset while Figure 8 shows its corresponding training time. As we can find from Figure 7, the testing rate generally increases when the number of ensembles is increasing for both two algorithms. However, the increment is very small when the

number of ensemble is larger than 15. It is readily to see that the improvements from using 1 ensemble to using 15 ensembles are around 2.5% and 3.5% for V-ELM and VOP-ELM, respectively. However, when further increasing the number of ensembles from 15 to 51, the improvements of 51 ensembles over 15 ensembles are only around 0.4% and 0.5%, respectively. But the training increases dramatically, especially for the VOP-ELM method. As shown in Figure 8, the training cost by VOP-ELM with 51 ensembles is longer than 1500 seconds. In addition, the training time used for both the two methods is proportional to the training time by the original ELM and OP-ELM, respectively. The proportional rate is related to the number of ensembles. Likewise, we can obtain the similar performance when using randomly generated training protein sequence dataset. Hence, considering both the enhancement of the classification rate and the increment of the training time, we suggest that the best choice of the number ensembles for both V-ELM and VOP-ELM should be less than 15.

5. Discussions and Conclusions

As demonstrated by the experimental results in Section 4, the proposed ensemble based VOP-ELM has the highest classification rate and outperforms V-ELM, OP-ELM, SVM, BP, and the original ELM algorithms for the protein sequence recognition. However, employing OP-ELM for each ensemble also increases the training time of the VOP-ELM method, as expected. Even though, VOP-ELM still learns much faster than the conventional BP and the popular SVM.

The original ELM has the fastest training speed of the SLFN model for the protein sequence classification problem among all classifiers. However, the classification performance on protein sequence by ELM is only comparable to the BP method and worse than the proposed ensemble based VOP-ELM and V-ELM, the OP-ELM, and SVM, in general. In addition, using the linear kernel can reduce the training time for some algorithms, such as the OP-ELM and the VOP-ELM algorithms. But the classification rate also reduces when employing the linear kernel as activation function. For the rest of algorithms, the training time used by the linear function is close to the one with the nonlinear kernels, as shown in Tables 2 and 4, respectively. Hence, how to choose the kernel function depends on the requirement of the protein sequence applications.

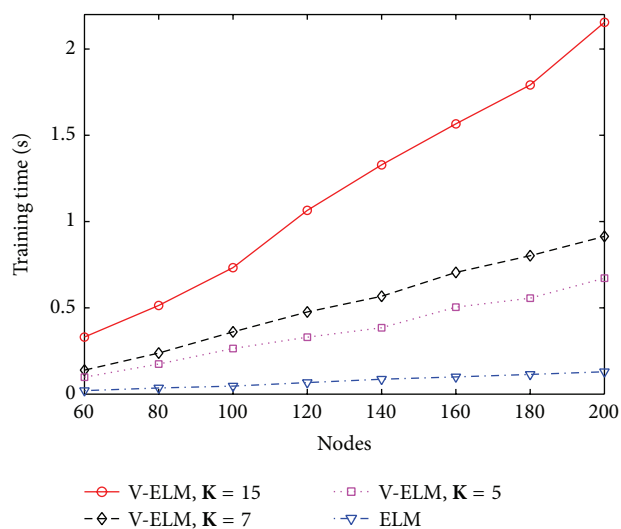


FIGURE 6: The training time of ELM and V-ELM w.s.t. different nodes on randomly generated training and testing protein sequence datasets.

Employing ELM as the training algorithm for each ensemble, the developed V-ELM outperforms the original ELM in general. The increment of V-ELM compared to ELM is more than 1% using the fixed protein sequence training dataset and is more than 2% using the randomly generated protein sequence training dataset for all the 3 kernels, respectively. Although the training time by V-ELM is longer than the one by ELM, the training phase for the protein sequence dataset still can be finished less than 1 second. Beside ELM, V-ELM can learn many times faster than the rest of compared algorithms. For certain applications, such as developing the online training model for protein sequence analysis, the original ELM and the V-ELM algorithm would be the best choices as researchers normally do a batch model retraining by using the past dataset combined with the new coming samples.

Therefore, as stated above, how to choose the proper classifier for the protein sequence classification depends on the requirements of the application. If only requiring the high recognition rate, the proposed VOP-ELM is the best choice. If the learning speed of the model for the protein sequence is the only concern, the original ELM is the proper classifier. However, if both the high classification rate and the fast training speed are required, the developed V-ELM and the existed OP-ELM should be used.

In conclusion, we have studied the protein sequence classification problem based on SLFNs in this paper. The existed OP-ELM has been first employed as the classifier. Then, the ensemble based structure of SLFNs has been proposed to enhance the performance of protein sequence classification. Two algorithms, namely, V-ELM and VOP-ELM, have been developed for protein sequence classifications by employing the original ELM and the OP-ELM to train each ensemble, respectively. Experimental results on the protein sequence datasets from the Protein Information Resource center demonstrated that VOP-ELM has the highest

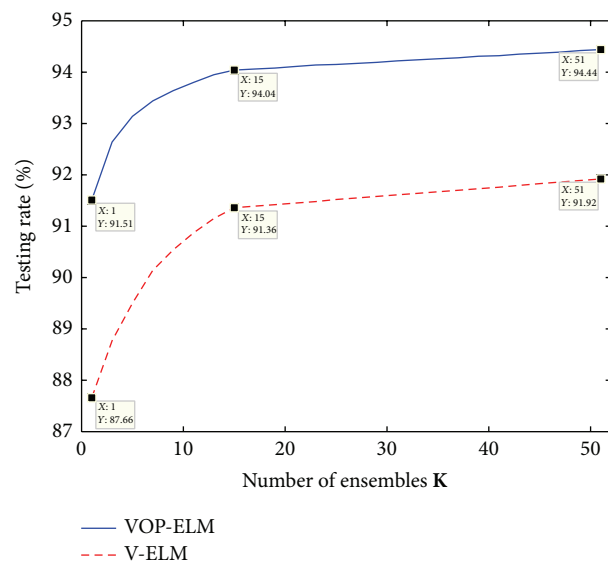


FIGURE 7: The testing rates of VOP-ELM and V-ELM w.s.t. different numbers of ensembles.

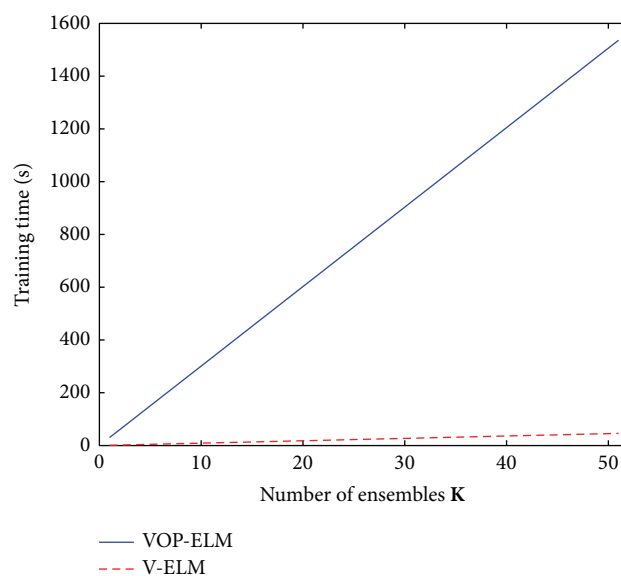


FIGURE 8: The training time of VOP-ELM and V-ELM w.s.t. different numbers of ensembles.

recognition rate among all compared state-of-art methods while the V-ELM outperforms the original ELM and BP but maintains the training speed as comparable as the original ELM. Moreover, to obtain a reasonable testing rate with an acceptable training time for the ensemble based algorithms, we have shown by simulations that the proper number of ensemble should be chosen less than 15.

Conflict of Interests

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

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China under Grant 61333009 and in part by the National Basic Research Program of China under Grant 2012CB821200.

References

- [1] A. Barve, S. Ghaskadbi, and S. Ghaskadbi, "Structural and sequence similarities of hydra xeroderma pigmentosum a protein to human homolog suggest early evolution and conservation," *BioMed Research International*, vol. 2013, Article ID 854745, 9 pages, 2013.
- [2] C. Chen, P. B. McGarvey, H. Huang, and C. H. Wu, "Protein bioinformatics infrastructure for the integration and analysis of multiple high-throughput omics data," *Advances in Bioinformatics*, vol. 2010, Article ID 423589, 2010.
- [3] H. Cong, M. Zhang, Q. Zhang et al., "Analysis of structures and epitopes of surface antigen glycoproteins expressed in bradyzoites of *Toxoplasma gondii*," *BioMed Research International*, vol. 2013, Article ID 165342, 9 pages, 2013.
- [4] J. Machado, A. C. Costa, and M. Quelhas, "Can power laws help us understand gene and proteome information?" *Advances in Mathematical Physics*, vol. 2013, Article ID 917153, 10 pages, 2013.
- [5] V. Carregari, R. Floriano, L. Rodrigues-Simioni et al., "Biochemical, pharmacological, and structural characterization of new basic PLA₂ bbil-tx from *Bothriopsis bilineata* snake venom," *BioMed Research International*, vol. 2013, Article ID 612649, 12 pages, 2013.
- [6] L. Liu, J. Cui, X. Zhang, T. Wei, P. Jiang, and Z. Wang, "Analysis of structures, functions, and epitopes of cysteine protease from *Spirometra erinacei* spargana," *BioMed Research International*, vol. 2013, Article ID 198250, 7 pages, 2013.
- [7] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach*, The MIT Press, Cambridge, Mass, USA, 2001.
- [8] Y. Yang, B.-L. Lu, and W.-Y. Yang, "Classification of protein sequences based on word segmentation methods," in *Proceedings of the 6th Asia-Pacific Bioinformatics Conference (APBC '08)*, vol. 6, pp. 177–186, Imperial College Press, 2008.
- [9] C. Caragea, A. Silvescu, and P. Mitra, "Protein sequence classification using feature hashing," *Proteome Science*, vol. 10, no. 1, pp. 1–8, 2012.
- [10] D. Wang, N. K. Lee, T. S. Dillon, and N. J. Hoogenraad, "Protein sequences classification using radial basis function (RBF) neural networks," in *Proceedings of the 9th International Conference on Neural Information Processing*, vol. 2, pp. 764–768, 2002.
- [11] D. Wang, N. K. Lee, and T. S. Dillon, "Extraction and optimization of fuzzy protein sequences classification rules using GRBF neural networks," *Information Processing Letters and Reviews*, vol. 1, no. 1, pp. 53–559, 2003.
- [12] D. Wang and G.-B. Huang, "Protein sequence classification using extreme learning machine," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '05)*, vol. 3, pp. 1406–1411, Montreal, Canada, 2005.
- [13] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 985–990, 2004.
- [14] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [15] G.-B. Huang, Q.-Y. Zhu, K. Z. Mao, C.-K. Siew, P. Saratchandran, and N. Sundararajan, "Can threshold networks be trained directly?" *IEEE Transactions on Circuits and Systems II*, vol. 53, no. 3, pp. 187–191, 2006.
- [16] K. Levenberg, "A method for the solution of certain problems in least squares," *The Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.
- [17] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [18] J. Cao, Z. Lin, and G.-B. Huang, "Self-adaptive evolutionary extreme learning machine," *Neural Processing Letters*, vol. 36, no. 3, pp. 285–305, 2012.
- [19] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: optimally pruned extreme learning machine," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 158–162, 2010.
- [20] T. Simila and J. Tikka, "Multiresponse sparse regression with application to multidimensional scaling," in *Proceedings of 15th International Conference Artificial Neural Networks: Formal Models and Their Applications*, pp. 97–102, 2005.
- [21] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [22] S. Haykin, *Neural Networks, A Comprehensive Foundation*, Pearson Education, Upper Saddle River, NJ, USA, 2001.
- [23] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 42, no. 2, pp. 513–529, 2012.
- [24] D. Serre, *Matrices: Theory and Applications*, Springer, New York, NY, USA, 2002.
- [25] G.-B. Huang, M.-B. Li, L. Chen, and C.-K. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, no. 4–6, pp. 576–583, 2008.
- [26] N.-Y. Liang, P. Saratchandran, G.-B. Huang, and N. Sundararajan, "Classification of mental tasks from EEG signals using extreme learning machine," *International Journal of Neural Systems*, vol. 16, no. 1, pp. 29–38, 2006.
- [27] R. Zhang, G.-B. Huang, N. Sundararajan, and P. Saratchandran, "Multi-category classification using an extreme learning machine for microarray gene expression cancer diagnosis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 3, pp. 485–494, 2007.
- [28] J. Cao, Z. Lin, and G.-B. Huang, "Composite function wavelet neural networks with extreme learning machine," *Neurocomputing*, vol. 73, no. 7–9, pp. 1405–1416, 2010.
- [29] Q.-Y. Zhu, A.-K. Qin, P.-N. Suganthan, and G.-B. Huang, "Evolutionary extreme learning machine," *Pattern Recognition*, vol. 38, no. 10, pp. 1759–1763, 2005.
- [30] J. Cao, Z. Lin, and G.-B. Huang, "Composite function wavelet neural networks with differential evolution and extreme learning machine," *Neural Processing Letters*, vol. 33, no. 3, pp. 251–265, 2011.
- [31] G. Bontempi, M. Birattari, and H. Bersini, "Recursive lazy learning for modeling and control," in *Proceedings of the 10th European Conference on Machine Learning*, pp. 292–303, 1998.

- [32] R. Myers, *Classical and Modern Regression with Applications*, Duxbury Press, 2nd edition, 2000.
- [33] J. Cao, Z. Lin, G.-B. Huang, and N. Liu, "Voting based extreme learning machine," *Information Sciences*, vol. 185, no. 1, pp. 66–77, 2012.