Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2018) 26: 1685 – 1697

doi:10.3906/elk-1706-60

Optimum, projected, and regularized extreme learning machine methods with singular value decomposition and L_2-Tikhonov regularization

Mohanad Abd SHEHAB, Nihan KAHRAMAN*
Department of Electronics and Communication Engineering, Faculty of Electrical and Electronics Engineering, Yıldız Technical University, İstanbul, Turkey

Received: 05.06.2017 • **Accepted/Published Online:** 11.04.2018 • **Final Version:** 27.07.2018

Abstract: The theory and implementation of an extreme learning machine (ELM) have proved that it is a simple, efficient, and accurate machine learning methodology. In an ELM, the hidden nodes are randomly initiated and fixed without iterative tuning. However, the optimal hidden layer neuron number (L_{opt}) is the key to ELM generalization performance where initializing this number by trial and error is not reasonably satisfied. Optimizing the hidden layer size using the leave-one-out cross validation method is a costly approach. In this paper, a fast and reliable statistical approach called optimum ELM (OELM) was developed to determine the minimum hidden layer size that yields an optimum performance. Another improvement that exploits the advantages of orthogonal projections with singular value decomposition was proposed in order to tackle the problem of randomness and correlated features in the input data. This approach, named projected ELM (PELM), achieves more than 2% advance in average accuracy. The final contribution of this paper was implementing Tikhonov regularization in the form of the L_2-penalty with ELM (TRELM), which regularizes and improves the matrix computations utilizing the L-curve criterion and SVD. The L-curve, unlike iterative methods, can estimate the optimum regularization parameter by illustrating a curve with few points that represents the tradeoff between minimizing the training error and the residual of output weight. The proposed TRELM was tested in 3 different scenarios of data sizes: small, moderate, and big datasets. Due to the simplicity, robustness, and less time consumption of OELM and PELM, it is recommended to use them with small and even moderate amounts of data. TRELM demonstrated that when enhancing the ELM performance it is necessary to enlarge the size of hidden nodes (L). As a result, in big data, increasing L in TRELM is necessary, which concurrently leads to a better accuracy. Various well-known datasets and state-of-the-art learning approaches were compared with the proposed approaches.

Key words: Extreme learning machines, singular value decomposition, Tikhonov regularization, optimum hidden nodes, orthogonal projection

1. Introduction

Guang and his group reported that a single hidden layer feedforward neural network (SLFN) that has a sufficient number of hidden neurons, arbitrarily assigned input weights, and biases with almost any nonlinear activation function can universally approximate any continuous functions or any compact input sets with zero or randomly small error [1]. Recently, extreme learning machine (ELM) has attracted a large number of researchers and engineers due to its rapidity and significant generalization performance. It has other merits including least human intervention, high learning efficiency, and fast learning speed [2–4].

*Correspondence: nicoskun@yildiz.edu.tr

ELM is a successful algorithm for both classification and regression [1–6], also extending the ability to include clustering [7]. It can be used efficiently with online sequential [8] as well as self-adaptive evolutionary algorithms [9], and in handling big data [10,11]. Almost all ELM studies have revealed that choosing the number of neurons in the hidden layer (L) is the key factor in determining the overall SLFN network architecture and performance. An appropriate method for the selection of the optimum number of hidden nodes is still unknown; it is set by the user and usually adjusted by trial and error [12–14]. To tackle this issue, researchers have proposed many ELM models that can properly select the network topology according to certain criteria. These models can be divided into 2 main categories: constructive (growing) methods such as incremental ELM [15,16] and error minimized ELM [17], and destructive (pruning) methods like pruned ELM [18] and optimally pruned ELM [6]. All the aforementioned methods have a serious shortcoming. They initiate a starting network structure and then gradually adjust the network depending on the error metric. They usually converge progressively with the lack of how to determine the starting network topology. The main objective of this paper is to develop a fast statistical algorithm that can find an appropriate network structure for an ELM and its variants without iteratively solutions.

2. ELM theoretical preliminaries

According to ELM theory [1], any nonlinear piecewise continuous activation function, G(.), can be used for feature mapping to approximate any continuous target function. Some examples of these functions are sigmoid, tangential, Gaussian, hinging, and ridge polynomials functions. In an ELM, the random choice of input layer weights and biases may improve the generalization properties of the solution of the linear output layer but does not guarantee producing valuable hidden layer features. Furthermore, in each round of simulation, the ELM solution fluctuates due to the random parameters of the hidden layer. A speedy and stable approach with singular value decomposition (SVD) was developed in order to exploit its orthogonal projections that can handle the discriminative feature data only.

The matrix solution of the standard ELM method may be close to singular and its pseudoinverse is prone to numerical instabilities; as a result, a small regularization term (λ) should be included to yield a regularization model of ELM (RELM). RELM tends to decrease prediction error and reduce the overfitting. Consequently, we have proposed a RELM version based on L_2-Tikhonov regularization and the L-curve to improve the robustness of the matrix computations and hence make the accuracy of classification and the RMSE of regression more reliable.

3. The proposed ELM method

3.1. Standard ELM with optimum hidden nodes (OELM)

According to investigations in various neural network studies, the number of neurons in the hidden layer (L) has an important relationship with the following parameters: the number of input features (n) and output (targets/classes) nodes (m), the amount and complexity of training data available, and the extent of accuracy and hence generalized error (ε). The proposed solution in determining L_{opt} is to examine a statistics-based approach (least square regression) combined with the previous factors as follows:

$$L_{opt} = \begin{cases} \alpha.(2n+m) & \text{for classification} \\ \alpha.(2n+1) & \text{for regression} \end{cases}, \tag{1}$$

where

$$\alpha = \begin{cases} 2^{\varepsilon} \text{ all regression coefficients} \neq 0 \\ 2^{\varepsilon+1} \text{ otherwise} \end{cases}, \tag{2}$$

where ε is the root mean square error that influences the target-dependent variable t_i.

To calculate the error ε, the model can take the regression coefficients (γ) form as:

$$t_i{=}x_i^T\gamma{+}\varepsilon_i, \quad i{=}\,1,\cdots,N. \tag{3}$$

Solving the least square problem LS with

$$\hat{\gamma} = (X^T X)^{-1} X^T t_i, \tag{4}$$

the predicted response is:

$$y_i{=}\hat{t}_i = X.\hat{\gamma}. \tag{5}$$

The residual mean errors are $\varepsilon_i^2{=}(t_i{-}y_i)^2$ and hence RMSE is:

$$\varepsilon{=}\sqrt{\sum_{i=1}^{N}\varepsilon_i^2}. \tag{6}$$

α depends on data size and modeling complexity type, i.e. $\alpha = 2^{\varepsilon}$, and this value is extended to $2^{\varepsilon+1}$ when one or more regression coefficients are zero, i.e. $\gamma_i{=}0$. This case will increase the complexity of handling data and hence increasing the hidden nodes is necessary. Although the input data distribution may not be linear, it is an essential step to calculate ε in formulating the linear regression model to simulate the least squares that solve the output weights $\beta{=}H^{\dagger}T$ of the ELM, which is a significant key for an ELM.

3.2. Projected ELM (PELM)

ELM models, like other machine learning approaches, tend to have problems when irrelevant, independent, or weakly correlated variables are present in the training dataset, and the randomly initialized weights can affect the performance and model generalization ability. Instead of choosing random weight vectors for w, the normalized constrained hidden weights were selected based on input sample distributions [19]. The only shortcoming in the constrained method is no guarantee of choosing the proper constrained vector distribution for all input samples.

The SVD matrix factorization principle can be employed to solve the above problems by generating orthogonal subspaces and eigenvalue base matrices and extracting useful features with a dimensionality reduction [20,21]. To suit nonsquare matrices, any matrix may be decomposed into a set of characteristic eigenvector pairs as $X^{m\times n}{=}UDV^T$, where $U^{m\times m}$ and $V^{n\times n}$ are orthogonal matrices and $D^{m\times n}$ is a diagonal matrix with singular values. As the ELM structure has only one hidden layer, the projection is applied to the input layer between the weight vector (w) and input attributes (X) [13,22]. In our proposed method, different forms of input weights are adopted based on input data (X) and the orthogonal principle available with SVD as follows: $w_1{=}V\sqrt{D^T}$,$w_2{=}VD^TU$,$w_3{=}VD^TU^T$,$w_4{=}V\sqrt{D^T}U$, and $w_5{=}V\sqrt{D^T}U^T$. After that, the eigenvalues of $(w_iX{+}b)$ and the ratio between maximum and minimum eigenvalues are calculated for all weights (w_i). It was

found empirically that the efficient form of the weight vector (w_i) yielded the largest ratio of the eigenvalues. The largest span between maximum and minimum eigenvalues can ensure generating discriminative feature mapping, which can approximate the desired function distinctly. It makes $H^{\dagger}$ nonsingular, stable, and square with a minimum row or column dimension that allows for fast calculations.

The proposed OELM and PELM algorithms are described here:

Consider a dataset containing N samples given as $(x_i,t_i)|x_i \in R^n, t_i \in R^m, i = 1, \cdots, N$, with n the number of input attributes and m the number of output classes, i.e. $X \in R^{N \times n}$ and $T \in R^{N \times m}$ with any suitable activation function $G(x)$. Using X and t_i, we find $\hat{\gamma}=(X^T X)^{-1} X^T t_i$ and then calculate the fitted response $y_i = X.\hat{\gamma}$ and the root mean error as $\varepsilon = \sqrt{\sum_{i=1}^{N} (t_i - y_i)^2}$, followed by using Eq. (2) to specify α.

1. Eq. (1) will then be used to calculate the best minimum number of hidden neurons L_{opt}
2. Depending on either randomly assigned input weight vectors w_i and hidden nodes bias b_i, $i = 1, \cdots, L_{opt}$ as used in OELM or the SVD projection principle as used in PELM:
   a. Find the singular value decomposition of input data (X), $UDV^T = SVD(X)$.
   b. Construct many sets of input weights as $w_1 = V\sqrt{D^T}, w_2 = VD^T U, w_3 = VD^T U^T, w_4 = V\sqrt{D^T}U$, and $w_5 = V\sqrt{D^T}U^T$.
   c. Calculate the eigenvalues of $(w_i X + b)$, $?i$, and choose the corresponding w_i that produces the largest ratio between the max and min eigenvalues.
   d. Choose $w = w(:, 1 : L_{opt})$ and the bias as $b = rand(L_{opt}, 1)$
3. Calculate the hidden layer output matrix

$$H(w_1, \cdots, w_{L_{opt}}, b_1, \cdots, b_{L_{opt}}, x_1, \cdots, x_N) = \begin{bmatrix} G(w_1,b_1,x_1) & \cdots & G(w_L,b_L,x_1) \\ \vdots & \cdots & \vdots \\ G(w_1,b_1,x_N) & \cdots & G(w_L,b_L,x_N) \end{bmatrix}_{NxL;L=L_{opt}}.$$

4. Extend the target vector t_i to $T_{ij} = \begin{cases} 1 \text{ for vectorsof class}_i = j \\ 0 \text{ for vectorsof class}_i \neq j \end{cases}$.
5. Calculate the hidden output weight $\beta : \beta = H^{\dagger} T$, where $H^{\dagger} = (H^T H)^{-1} H^T$ is the pseudoinverse of hidden layer output matrix H.

   From the perspective of evaluation, the samples are divided into training and testing sets. The training sets are adopted first to obtain the value of the output weight (β), and then:
6. β is used to fit or classify the test patterns of the output label (y_{test}) using

$$y_{test} = arg_{max}^{row}(H_{test}\beta).$$

3.3. Tikhonov regularization with L-curve

Tikhonov regularization (TR), in statistics, is known as ridge regression that allows the addition of a small positive value called the regularization parameter (λ) to the diagonal $\mathbf{H^T H}$ or $\mathbf{HH^T}$ to gain more stability, a robust solution, and good generalization performance, thus avoiding model overfitting and improving overall prediction accuracy. It is probably the most successful regularization method of all time [23]. Different approaches for calculating the proper regularization parameter (λ) have been employed, such as the discrepancy principle, generalized cross validation (GCV), and the L-curve [24]. Appropriate model selection and parameter optimization can be achieved with leave-one-out cross validation (LOOCV), which consumes a large amount of computations [6,10,25,26], an adopted efficient computing method depending on the prediction sum of squares (PRESS) formula to calculate the cross validation minimum square error ($\mathbf{MSE_{CV}}$) utilizing Eq. (7):

$$\mathbf{MSE_{CV(K)}}=\frac{\mathbf{1}}{\mathbf{K}}\sum_{\mathbf{i=1}}^{\mathbf{K}}\left(\frac{\mathbf{t_i}-\hat{y}_\mathbf{i}}{\mathbf{1}-(\mathbf{HAT_{ii}})}\right)^{\mathbf{2}}, \tag{7}$$

where

$$HAT = HH^{\dagger} = H.(H^T H)^{-1} H^T. \tag{8}$$

In PRESS, estimating the pseudoinverse and $\mathbf{MSE_{CV}}$ by $\mathbf{K}$-repetitions would be computationally expensive, leading to missing the advantage of the ELM in fast predicting. Like the GCV approach, the L-curve method does not depend on specific or prior knowledge of the noise variance. As noted, the difficulty in GCV is that its function can have a very flat minimum, making it difficult to determine the optimal λ numerically. On the other hand, the GCV rule may be unstable for correlated noise, resulting in undersmoothing [25].

The L-curve is a log-log plot that seeks to determine a proper value of λ that balances between 2 error components, training error versus the residual norm of output weight, while keeping our objective function as in Eq. (9):

$$\mathbf{min}_{\lambda,\beta}\{\,\|\mathbf{H}\beta-\mathbf{T}\|_\mathbf{2}^\mathbf{2}+\lambda\,\|\beta\|_\mathbf{2}^\mathbf{2}\}. \tag{9}$$

The L-curve method consists of the analysis of the piecewise linear curve, whose breakpoints are:

$$(\mathbf{x_i},\mathbf{y_i})=(\log\|\mathbf{H}\beta_\mathbf{i}-\mathbf{T}\|_\mathbf{2}^\mathbf{2}\,\mathbf{log}\lambda\,\|\beta_\mathbf{i}\|_\mathbf{2}^\mathbf{2}),\quad \mathbf{i=1\ \ top}, \tag{10}$$

where $\mathbf{p}$ is the row dimension of the regularization matrix, which varies depending on the resolution level.

The L-curve basically consists of 2 parts: a "flat" part where the regularization errors dominate and a "steep" part where the perturbation error dominates. This curve in most cases exhibits a typical L shape, and the optimal value of the regularization parameter λ must lie on the corner of the L. The L-curve is usually more amenable numerically due to its simplicity, accuracy, ability to deal with large scale matrices, and cost that is less than or similar to other regularization methods [27]. Incorporation of the regularization parameter and various recombinations of $\mathbf{H_i}$ in the SVD method with a suitable λ extraction approach (L-curve) was used to obtain the various $\mathbf{H}^\dagger$s with minimal recomputation and hence the optimum $\lambda_\mathbf{opt}\mathbf{and}\beta_\mathbf{opt}$.

Due to the variety of experimental data types, the proposed algorithm can use different subsets H_i depending on random permutation or limited LOOCV as a valuable approach to overcome the overfitting or if the data samples are small by the bootstrapping method.

Suppose a training set of N samples is given as $\{(x_i, t_i)|x_i \in R^n, t_i \in R^m, i = 1, \cdots, N\}$, with any suitable activation function, using $X \in R^{N\times n}$, $t \in R^N$, and the hidden layer output and target matrices $H_{N\times L} and T_{N\times m}$. Then permute H into different max random subsets H_i, i.e. $H_i \epsilon H \Longrightarrow \{H_1 H_2, \ldots, H_{max}$.

Employ H_i with SVD matrix factorization and L-corner curve methods as follows:

1. For $i = 1 : max$, we used $max = 20$.
2. Take H_i and T utilizing the L-curve regularization method by employing the Hansen Regularization Toolbox Package [28] to find λ_i corresponding to H_i that satisfies $min_{\lambda_i}\{\|H_i\beta - T\|_2^2 + \lambda_i \|\beta\|_2^2\}$.
3. End.
4. Select the minimum λ_i as λ_{opt} and corresponding H_i as the usable H; they lead to β_{opt}.
5. Based on SVD factorization:

   (a) $H^T H$ if $N \geq L \rightarrow SVD(H^T H) = VD^2V^T$,

   (b) HH^T if $L \geq N \rightarrow SVD(HH^T) = UD^2U^T$. (11)

6. Calculate β_{opt} and hence $y_{opt=H.\beta_{opt}}$ using the Woodbury formula [29] as:

   (a) for $N \geq L$: $\beta_{opt} = (H^T H + \lambda_{opt} I_L)^{-1} H^T T = V(D^2 + \lambda_{opt} I_L)^{-1} V^T H^T T =$

$$\sum_{i=1}^{N} v_i \left(\frac{1}{d_{ii}^2 + \lambda_{opt}}\right) v_i^T . H^T T, \tag{12}$$

   (b) for $L \geq N$: $\beta_{opt} = H^T (HH^T + \lambda_{opt} I_N)^{-1} T = H^T U(D^2 + \lambda_{opt} I_N)^{-1} U^T T =$

$$H^T \sum_{i=1}^{N} u_i \left(\frac{1}{d_{ii}^2 + \lambda_{opt}}\right) u_i^T . T. \tag{13}$$

4. Experimental results

The ELM was tested with 3 situations: optimum hidden number, the SVD projection, and TR regularization. The experimental results and runtime reported were based on the average of 20 independent trials for regularly sized datasets. All simulations were implemented using the MATLAB 8.1 (R2013a) environment and performed on an Intel Core i5, 2.4 GHz CPU, 4 GB RAM computer. Eleven datasets from the University of California at Irvine (UCI) Machine Learning Repository [30] were tested: 4 for classification and 7 for regression with 20 different random permutations taken without replacement. Two-thirds of the samples were used for training and one-third for testing. Big data were handled with 2 extra datasets [31]: the AR Face Database with 700 $\times$ 300 training and 700 $\times$ 300 testing samples has 100 individual face recognitions, and the USPS (US Postal Service) Digits Database consists of handwritten digits from 0 to 9 with 7291 $\times$ 256 training and 2007 $\times$ 256 testing samples. The results showed that OELM, PELM, and TRELM can predict or fit the desired targets rapidly at low error rates even with large datasets.

4.1. Standard ELM with optimum hidden nodes (OELM)

The performance (accuracy and RMSE ± standard deviation over the computational time) of the primary ELM learning algorithm was examined and elaborated first with optimum hidden node number (L_{opt}) in Eqs. (1) and (2) on the 11 UCI datasets as presented in Tables 1 and 2 and Figures 1a, 1b, 2a, and 2b. The simplest datasets like "Diabetes and Wine" in Table 1" and "Stock and Breast Cancer" in Table 2 are of root mean square error $\varepsilon \approx 0, so \alpha \approx 1$ and hence the optimum L_{opt} will be $(2n+m)$ where $m=1$ for regression. Figures 1a and 2a show the relationship between hidden nodes number with average testing accuracy where each value of accuracy represents the average of 20 different random permutations. Figures 1b and 2b estimate the optimal hidden node size using k-fold cross validation with k = 10 inside the testing set. Here, cross validation with Figures 1b and 2b is employed to validate the ability of OELM in finding the optimum (L_{opt}).

Table 1. OELM performance results for classification samples.

Dataset/types	n/m	α	L_{opt}	Train. data	Test. data	Train. accuracy (%) / time (s)	Testing accuracy (%) / time (s)
P. I. Diabetes/*	8/2	1.32	23	576	192	78.95 ± 1.23 / 0.007	77.34 ± 2.86 / 0.002
Wine/*	13/3	1.20	34	115	63	99.52 ± 0.66 / 0.0044	96.04 ± 1.89 / 0.003
Segment/**	19/7	4.31	194	1500	810	96.27 ± 0.3 / 0.15	94.77 ± 0.57 / 0.039
Satellite/**	36/7	2.32	183	4435	2000	93.97 ± 0.49 / 0.35	89.44 ± 0.58 / 0.065

*Small data type, **moderate data type.

Table 2. OELM performance results for regression samples.

Dataset/type	n	α	L_{opt}	Train. data	Test. data	Training RMSE / time (s)	Testing RMSE / time (s)
Stock/*	9	1.05	20	634	316	0.0993 ± 0.0118 / 0.002	0.0934 ± 0.009 / 4.0e-04
Breast Cancer/*	32	1.07	69	130	64	0.0983 ± 0.0026 / 0.007	0.103 ± 0.005 / 3.8e-04
Bank/*	8	1.01	17	3000	1500	0.012 ± 2.36e-04 / 0.017	0.012 ± 3.6e-04 / 0.004
Ailerons/**	40	2	162	4770	2384	2.0e-05 ± 2.9e-06 / 0.06	2.0e-05 ± 0.2e-06 / 0.005
D_Elevators/**	6	2.3	30	6345	3172	1.7e-04 ± 2.1e-05 / 0.04	1.8e-04 ± 2.3e-05 / 0.01
Elevators/**	18	2	74	5835	2917	1.0e-04 ± 1.8e-06 / 0.13	0.8e-04 ± 2.7e-6 / 0.02
Kinematics/**	8	2.9	49	5462	2730	0.202 ± 0.0093 / 0.017	0.204 ± 0.0086 / 0.0057

Figure 1 is for the 'Pima Indians Diabetes' classification dataset and it is clear that the best testing accuracy is at $L_{opt} = 20$ from the average test and $L_{opt} = 23$ from the LOOCV test. Figure 2 is for the 'Segment' dataset, which produces the highest accuracy with lower optimum hidden node number $L_{opt} = 200$. The L_{opt} values obtained from different approaches are comparable, which demonstrates the validity of the proposed method.

4.2. Projected ELM (PELM)

PELM was tested on the same databases (classification type only) and the results are reported in Table 3. The simulation results are in line with the earlier standard ELM, which confirmed the increase in the accuracies (about 2% or more) with or without a slight increase in training time. Moreover, the standard deviations for

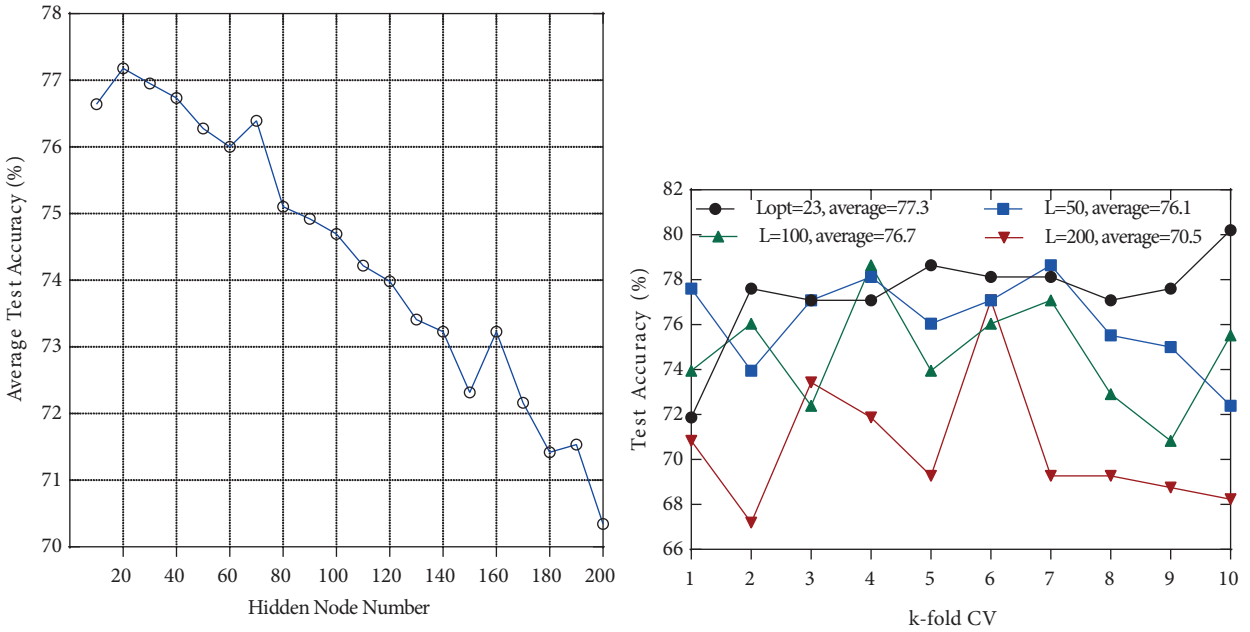

Figure 1. The relationship of testing accuracy with (a) hidden nodes number and (b) k-fold number for Pima Indians Diabetes dataset.

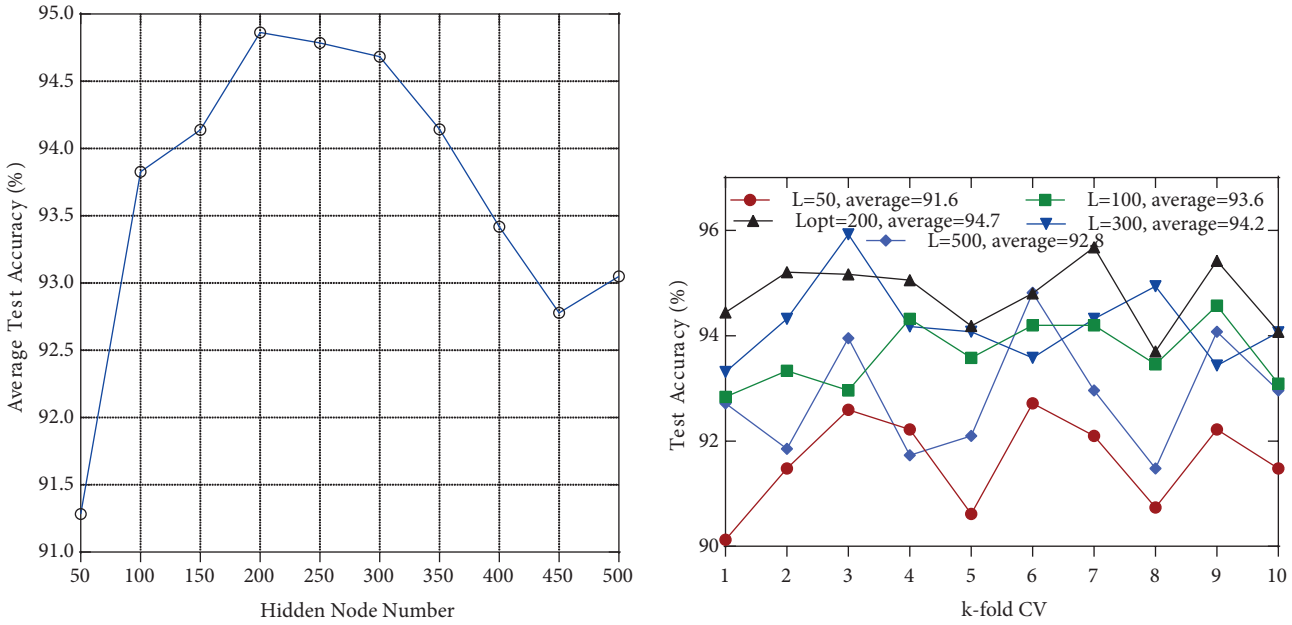

Figure 2. Testing accuracy relation with (a) L and (b) k-fold number for Segment dataset.

all datasets dropped significantly, which means that PELM is more stable and robust against variations. In order to decrease the required time in PELM, it can exploit the SVD as a dimensionality reduction method to reduce the applied input data $\mathbf{X}\in\mathbf{R}^{\mathbf{N}\times\mathbf{n}}$ to lower dimensionality spaces $\in\mathbf{R}^{\mathbf{N}\times\mathbf{d}}$, i.e. $\mathbf{d}<\mathbf{n}$, and hence improve the overall performance.

Table 3. PELM performance results.

Dataset/type	n/m	α	L_{opt}	Training data	Testing data	Train. accuracy (%) / time (s)	Test. accuracy (%) / time (s)
P.I. Diabetes/*	8/2	1.32	23	576	192	79.91 ± 0.9 / 0.0063	78.7 ± 2.12 / 0.002
Wine/*	13/3	1.20	34	115	63	99.7 ± 0.39 / 0.0047	97.62 ± 1.42 / 0.0045
Segment/**	19/7	4.31	194	1500	810	97.06 ± 0.17 / 0.165	96.81 ± 0.4 / 0.04
Satellite/**	36/7	2.32	183	4435	2000	95.38 ± 0.37 / 2.145	91.73 ± 0.47 / 0.1709

4.3. L_2-Tikhonov regularization ELM with L-curve (TRELM)

For an ELM, choosing an appropriate or optimum regularization parameter (λ_{opt}) with good efficiency and speed is crucial. As can be observed from Figure 3 for the segment image database, the best regularization parameter is λ_{opt}= 0.02817, which can be calculated from the corner of the L-curve that tries to suppress as much as possible the influence of both error norms at the same time. Figure 3 reveals that, in spite of the large input data space of the segment image, the construction of the L-curve is made of 10 points only, i.e. p= 10 row dimension. The p-dimension is flexible and depends on the degree of regularization precision. The advantages of using the L-curve include the easiness and speed to present the set of the 2 axis points; it is also flexible in choosing the number of represented points. The performance of the proposed TRELM model was tested for all types of data sizes (small, moderate, and large) for both classification and regression problems as stated in Tables 4 and 5. These tables reveal that TRELM is applicable, robust, and of optimum solutions for both moderate and large data experiments. TRELM can tackle a wide range of machine learning problems that yield lower error rates with large L in classification and normal L in regression. Moreover, for the small dataset, the TRELM accuracy/RMSE is comparable to the OELM approach but with longer time, so TRELM is not recommended for small datasets. For all data types, it is obvious that the ELM with regularization needs more time than that without it because it operates properly with many hidden nodes; also, TRELM has more stable response, which means it has lower variations than OELM and PELM. For the AR and USPS datasets, Figures 4a and 4b show the average accuracy comparison over many runs among the 3 proposed approaches (OELM, PELM, and TRELM). It is apparent that OELM and PELM do not have a linear relationship as in TRELM with respect to the number of hidden nodes.

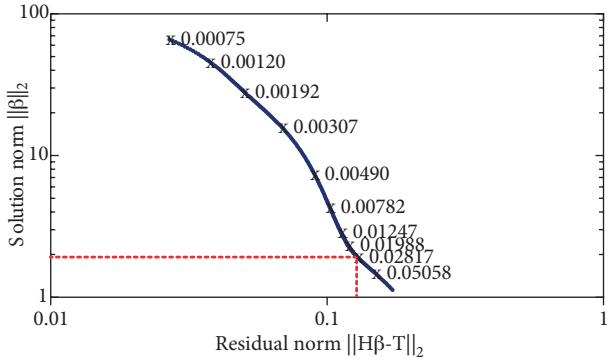

Figure 3. L-curve regularization method for Segment image database for λ_{opt}.

Many studies [1–18] demonstrated the superiority and effectiveness of the ELM and its improved varieties

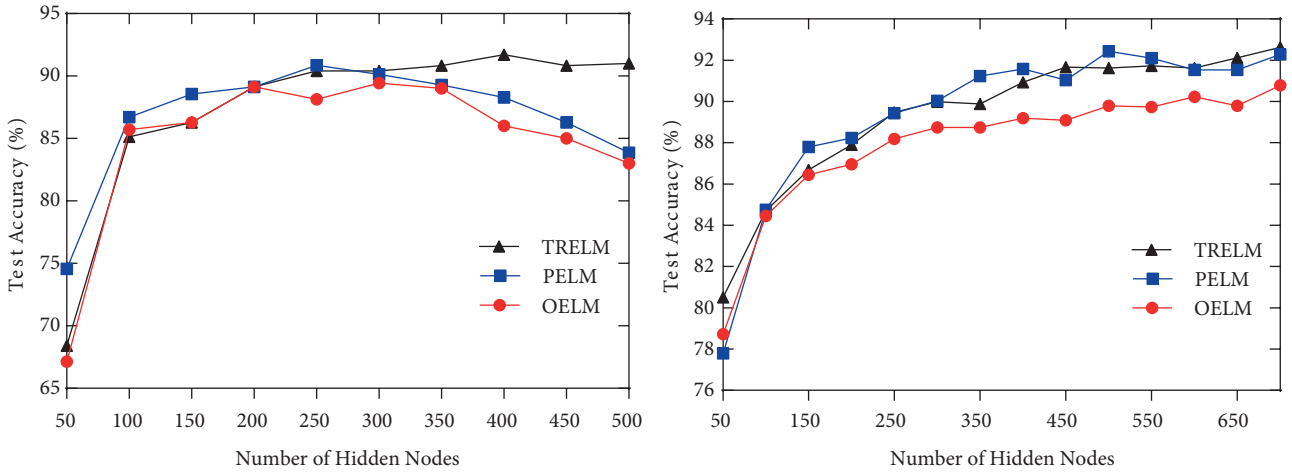

Figure 4. ELM models accuracy for (a) AR and (b) USPS datasets.

Table 4. TRELM performance results for classification datasets.

Dataset	Type	λ_{opt}	L	Train. acc. / time (s)	Test. acc. / time (s)
P. I. Diabetes	Small	0.1108	1300	81.4 ± 1.107 / 0.7970	75.5 ± 1.97 / 0.0058
Wine	Small	0.0183	2100	97.4 ± 0.48 / 0.0204	98.41 ± 1.38 / 0.0033
Segment	Moderate	0.0281	1100	97.1 ± 0.19 / 3.3280	95.43 ± 0.43 / 0.0289
Satellite	Moderate	0.4493	1900	94 ± 0.31 / 19.6933	90.1 ± 0.37 / 0.2919
AR	Large	0.2466	2100	100 ± 0.27 / 1.5643	93.78 ± 0.36 / 0.0712
USPS	Large	54.598	2100	99.7 ± 0.73 / 34.67	94.94 ± 0.91 / 0.1733

Table 5. TRELM performance results for regression datasets.

Dataset/type	λ_{opt}	L	Train. RMSE / time (s)	Test. RMSE / time (s)
Stock/*	0.00034	250	0.0459 ± 0.0011 / 0.1535	0.0540±0.0025 / 0.0026
Breast Cancer/*	variable	> 100	0.0982 ± 0.0060 / 0.0279	0.1078±0.0078 / 0.0062
Ailerons/**	0.06081	150	8.16e-06 ± 5.96e-07 / 0.7541	8.4e-06 ± 6.44e-07 / 0.0141
Bank/**	0.00248	120	6.448e-04 ± 6.90e-05 / 0.384	0.002 ± 1.486e-04 / 0.0058
D-Elevators/**	0.04979	120	1.39e-04 ± 1.82e-06 / 0.8191	1.40e-04 ± 2.42e-06 / 0.0133
Elevators/**	0.02237	260	9.68e-05 ± 1.41e-06 / 1.589	9.88e-5 ± 2.90e-06 / 0.0258
Kinematics/**	0.07427	800	0.0952 ± 0.0022 / 5.8260	0.1134 ± 0.0038 / 0.0787

as the fastest nonlinear predication approach, better than or comparable to the generalization and performance of widely used state-of-the-art learning algorithms such as the nearest neighbor classifier (NN), linear discriminant analysis (LDA), linear and least square support vector machine (LSVM and LS_SVM) [32], logistic regression classifier (LR) [33], and collaborative representation-based regularized least square (CRC_RLS) [34]. For fair comparison, we compare the TRELM paradigm with the AR and USPS databases as in Tables 6 and 7. It is clear that TRELM has both speed and accuracy advantages over the abovementioned learning methods.

Table 6. Performance comparison for AR datasets.

Classifier method	NN	LDA	LSVM	LR	CRC_RLS	TRELM
Testing accuracy (%)	71.57	84.71	75.85	77.42	93.62	93.78
Testing time (s)	0.193	0.421	3.987	0.144	NA	0.071

Table 7. Performance comparison for USPS datasets.

Classifier method	NN	LDA	LSVM	LR	CRC_RLS	TRELM
Testing accuracy (%)	92.82	80.61	93.37	91.18	93.78	94.94
Testing time (s)	4.86	0.373	8.102	0.282	NA	0.1733

5. Conclusion and future works

In ELMs with fixed network architecture, the suitable number of hidden nodes (L) is the key to ELM performance and it is the only factor that needs to be tuned by the user. In most cases, L is arbitrarily initiated and then it is gradually increased or decreased by a fixed interval. A nearly optimal number is then selected based on the error metric with the cross validation method. It is, however, quite time-consuming. In this work, a simple and fast approach based on the least square method was developed for calculating the minimum optimum hidden node number (L_{opt}). As was assessed with the cross validation tool, our investigated approach can attain the desired L_{opt} or close to it for different data types with minimal user intervention.

In an ELM, both the randomness of the SLFN input weights and correlated features can result in poor generalization performance or ill-posed problems. To overcome these restrictions, SVD was exploited significantly as in PELM to produce efficient input features and enhance the parameter selections. For more improvement in recognition precision and speed, it can extend the use of SVD to reduce the dimensionality of the input data. Finally, L_2-regularized Tikhonov regularization was added to the ELM (TRELM) to prevent the model from overfitting and improve the overall robustness. The L-curve and SVD approaches were utilized implicitly within the TRELM to find the regularization parameter, to retain the fast ELM learning advantage, and to dramatically reduce the complexity of the matrix calculations. As long as L is large enough, TRELM is more accurate, is less sensitive to the changing of L, and needs a longer time than OELM and PELM. Now building a suitable ELM with optimum hidden nodes is easy and fast and has an automatic learning ability where the users have limited impact, so construction of suitable hardware capable of working with the optimum ELM for biometric classification applications is hereby recommended. It will be beneficial if other matrix factorizations and valuable regularizations like L_1 or L_{21} types can be included within the ELM models to handle outliers and imbalances in input data.

References

[1] Huang GB, Zhu QY, Siew CK. Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE International Joint Conference on Neural Networks; 2004. New York, NY, USA: IEEE. pp. 489-501.

[2] Wang Y, Cao F, Yuan Y. A study on effectiveness of extreme learning machine. Neurocomputing 2011; 74: 2483-2490.

[3] Luo M, Zhang K. A hybrid approach combining extreme learning machine and sparse representation for image classification. Eng Appl Artif Intel 2014; 27: 228-235.

[4] Huang GB, Song S, You K. Trends in extreme learning machines: a review. Neural Networks 2015; 61: 32-48.

[5] Liu X, Wang L, Huang GB, Zhang J, Yin J. Multiple kernel extreme learning machine. Neurocomputing 2015; 149: 253-264.

[6] Miche Y, Sorjamaa A, Bas P, Simula O, Jutten C, Lendasse A. OP-ELM: Optimally pruned extreme learning machine. IEEE T Neural Networ 2010; 21: 158-162.

[7] Huang G, Song S, Gupta J, Wu C. Semi-supervised and unsupervised extreme learning machines. IEEE T Cybernetics 2014; 44: 2405-2417.

[8] Liang NY, Huang GB, Saratchandran P, Sundararajan N. A fast and accurate online sequential learning algorithm for feedforward networks. IEEE T Neural Networ 2006; 7: 1411-1423.

[9] Cao J, Lin Z, Huang GB. Self-adaptive evolutionary extreme learning machine. Neural Process Lett 2012; 36: 285-305.

[10] Akusok A, Björk KM, Miche Y, Lendasse A. High-performance extreme learning machines: a complete toolbox for big data applications. IEEE Access 2015; 3: 1011-1025.

[11] Anton A. Extreme learning machines: novel extensions and application to big data. PhD, University of Iowa, Iowa City, Iowa, USA, 2016.

[12] Yang YM, Wang YN, Yuan XF. Bidirectional extreme learning machine for regression problem. IEEE T Neural Networ 2012; 23: 1498-1505.

[13] Cambria E, Huang GB. Extreme learning machines [trends & controversies]. IEEE Intell Syst 2013; 28: 30-59.

[14] Huang GB. What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John Neumann's puzzle. Cogn Comput 2015; 7: 263-278.

[15] Huang GB, Chen L, Siew CK. Universal approximation using incremental constructive feedforward networks. IEEE T Neural Networ 2006; 17: 879-892.

[16] Huang GB, Chen L. Enhanced random search based incremental extreme learning machine. Neurocomputing 2008; 71: 3460-3468.

[17] Feng G, Huang GB, Lin Q, Gay R. Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE T Neural Networ 2009; 20: 1342-1357.

[18] Rong HJ, Ong YS, Tan AH, and Zhu Z. A fast pruned-extreme learning machine for classification problem. Neurocomputing 2008; 72: 359-366.

[19] Zhu W, Miao J, Qing L. Constrained extreme learning machines: a study on classification cases. Journal of Computer Vision Pattern Recognition 2015; 14: 1-14.

[20] Moravec P and Snasel V. Dimension Reduction Methods for Iris Recognition. Spindleruv Mlyn, Czech Republic: Czech Technical University in Prague, 2009.

[21] Pisani D. Matrix decomposition algorithms for feature extraction. PhD, University of Malta, Msida, Malta, 2004.

[22] Xu X, Wang Z, Zhang X, Yan W, Deng W and Lu L. Human face recognition using multi-class projection extreme learning machine. In: IEIE Transactions on Smart Processing & Computing; 2013. pp. 323-331.

[23] Giovannelli JF, Idier J. Regularization and Bayesian Methods for Inverse Problems in Signal and Image Processing. London, UK: Wiley-ISTE, 2015.

[24] Chung J, Espanol MI, Nguyen T. Optimal regularization parameters for general form Tikhonov regularization. Inverse Probl 2014; 33: 1-21.

[25] Heeswijk M, Miche Y. Binary/ternary extreme learning machines. Neurocomputing 2015; 149; 187-197.

[26] Grigorievskiy A, Miche Y, Käpylä M, Lendasse A. Singular value decomposition update and its application to INC-OP-ELM. Neurocomputing 2016; 174: 99-108.

[27] Xiang H, Zou J. Regularization with randomized SVD for large-scale discrete inverse problems. Inverse Probl 2013; 29: 1-23.

[28] Hansen PC. Regularization Tools: A MATLAB Package for Analysis and Solution of Discrete Ill-Posed Problems. Version 4.1 for MATLAB 7.3. Natick, MA, USA: MathWorks, 2008.

[29] Petersen KB, Pedersen MS. The Matrix Cookbook. Waterloo, Canada: University of Waterloo, 2007.

[30] UCI. UCI Machine Learning Repository. Irvine, CA, USA: UCI, 2016.

[31] Cao J, Zhang K, Luo M, Yin C, Lai X. Extreme learning machine and adaptive sparse representation for image classification. Neural Networks 2016; 81: 91-102.

[32] Suykens JA, Vandewalle J. Least squares support vector machine classifiers. Neural Process Lett 1999; 9: 293-300.

[33] Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting. Ann Stat 2000; 28: 337-407.

[34] Zhang L, Yang M, Feng X. Sparse representation or collaborative representation: which helps face recognition. In: Proceedings of the IEEE International Conference on Computer Vision; 2011. New York, NY, USA: IEEE. pp. 471-478.