

Research Article

Improved Extreme Learning Machine and Its Application in Image Quality Assessment

Li Mao,¹ Lidong Zhang,¹ Xingyang Liu,¹ Chaofeng Li,¹ and Hong Yang²

¹ Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), School of Internet of Things, Jiangnan University, Wuxi, Jiangsu 214122, China

² Freshwater Fisheries Research Center, Chinese Academy of Fishery Science, Wuxi, Jiangsu 214081, China

Correspondence should be addressed to Chaofeng Li; wxlichaocheng@126.com

Received 11 February 2014; Revised 6 May 2014; Accepted 8 May 2014; Published 22 May 2014

Academic Editor: Swagatam Das

Copyright © 2014 Li Mao et al. 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.

Extreme learning machine (ELM) is a new class of single-hidden layer feedforward neural network (SLFN), which is simple in theory and fast in implementation. Zong et al. propose a weighted extreme learning machine for learning data with imbalanced class distribution, which maintains the advantages from original ELM. However, the current reported ELM and its improved version are only based on the empirical risk minimization principle, which may suffer from overfitting. To solve the overfitting troubles, in this paper, we incorporate the structural risk minimization principle into the (weighted) ELM, and propose a modified (weighted) extreme learning machine (M-ELM and M-WELM). Experimental results show that our proposed M-WELM outperforms the current reported extreme learning machine algorithm in image quality assessment.

1. Introduction

Extreme learning machine (ELM) was proposed as a new class of single-hidden layer feedforward neural network by Huang et al. [1]. Its basic idea is to set a suitable number of nodes in the hidden layer before training and randomly assign the values for the input weights and offsets of the hidden layer in the implementation procedure. The algorithm completes the whole process at once and generates a unique optimal solution without the necessity of iterations. So it has the advantages of easy parameter selection and fast learning speed. Liang et al. [2] also proposed an online sequential extreme learning machine algorithm (OS-ELM) that can learn data one-by-one or chunk-by-chunk. Although OS-ELM provides better generalization performance, it excessively depends on experimental data. Lan et al. [3] presented an ensemble of online sequential extreme learning machine (EOS-ELM), which is a more stable integrated network structure consisting of multiple OS-ELM networks. Rong et al. [4] developed an OS-Fuzzy-ELM algorithm by combining TSK fuzzy inference system and ELM algorithm, which reduces the training time significantly. Feng et al. [5] presented an

improved ELM algorithm based on error minimization. In [6], Zong et al. proposed a weighted ELM for dealing with data with imbalanced class distribution, which is able to be generalized to balanced data and maintains the advantages from original ELM. However all these algorithms only consider the empirical risk minimization principle, which can easily lead to overfitting [7].

Support vector machine (SVM), proposed by Cortes and Vapnik [8], is actually also a single-hidden layer feedforward network. In [9–11], Suykens et al. proposed the least-squares support vector machine (LS-SVM), which transforms the linear inequality constraints into linear equality constraints in the support vector machine and, thus, converts solving the QP problem into solving linear equations. It reduces the difficulty of support vector machine learning a great deal of samples and also improves the efficiency. Both the SVM and LS-SVM are general algorithms based on guaranteed risk bounds of statistical learning theory, that is, the so-called structural risk minimization (SRM) principle, which improves their generalization ability.

In this paper, to lower the overfitting phenomena of the extreme learning machine algorithms, we refer to the

LS-SVM algorithm, draw the structural risk minimization principle into the ELM and WELM algorithms, and propose a modified ELM and WELM algorithm and call them as M-ELM and M-WELM. Our experimental results suggest the validity of our proposed M-ELM and M-WELM algorithm.

The structure of this paper is organized as follows. The brief introduction to ELM and weighted ELM is done in Section 2. In Section 3, the principles of our proposed M-ELM and M-WELM are described. Experimental results and performance assessment are presented in Section 4. In Section 5, the conclusion is presented.

2. Brief Introduction to Extreme Learning Machine

2.1. Extreme Learning Machine (ELM). Extreme learning machine (ELM) proposed is a single-hidden layer feedforward networks (SLFNs) which randomly selected the input weights and analytically determines the output weights of SLFNs [1, 2, 12]. One key principle of the ELM is that one may randomly choose and fix the hidden node parameters. After the hidden nodes parameters are chosen randomly, SLFN becomes a linear system where the output weights of the network can be analytically determined using simple generalized inverse operation of the hidden layer output matrices [13].

For an observation data set with N nodes in the hidden layer and the excitation function G , the extreme learning machine model can be expressed as

$$f(x) = \sum_{i=1}^N \beta_i G(a_i, b_i, x_i) = \beta \cdot \mathbf{h}(x), \quad (1)$$

where β_i is the output weight of the i th hidden layer node and the output neuron, a_i is the input weight of the input neuron and the i th hidden layer node, and b_i is the offset of the i th hidden layer node. Consider $\mathbf{h}(x) = [G(a_1, b_1, x_1), \dots, G(a_N, b_N, x_N)]$ denotes the output matrix of hidden layer. a_i and b_i are randomly selected before training and remain the same in the training procedure. The output weights β_i can be obtained by solving the least-squares solutions of the following linear equation:

$$\min \sum_{i=1}^N \|\beta_i \cdot \mathbf{h}(x_i) - y_i\|. \quad (2)$$

The least-squares solution to the equations is

$$\beta = H^+ Y, \quad (3)$$

where H^+ is called the Moore-Penrose generalized inverse of the hidden layer output matrix H .

2.2. LS-SVM Regression. Assume that an input and output sample data set for regression analysis is $T = \{(x_1, y_1), \dots, (x_l, y_l)\}$, where $x_i \in R^n$ and $y \in R$, $i = 1, \dots, l$. LS-SVM regression algorithm maps the data X into a high-dimensional feature space F through a nonlinear mapping ϕ and does linear regression in the space F . The regression

estimation for the observation data set given above can be formulated as below, where ω and b^* are the regression factors,

$$f(x) = (\omega \cdot \phi(x)) + b^*. \quad (4)$$

LS-SVM regression method is used to solve the weight vector ω and deviation b^* . Based on the structural risk minimization, the optimization model of the optimal regression function [9–11] can be established as

$$\begin{aligned} \min \quad & \frac{1}{2} \|\omega\|^2 + \frac{1}{2} \gamma \sum_{i=1}^l \xi_i^2 \\ \text{s.t. (subject to)} \quad & y_i - f(x_i) = \xi_i \\ & \xi_i \geq 0, \quad i = 1, 2, \dots, l, \end{aligned} \quad (5)$$

where γ is the penalty constant, which is a compromise between complexity and fitting accuracy of regression model. Higher value γ means higher fitting degree. ξ_i is the slack variable. LS-SVM transforms the inequality constraints into equality constraints by defining loss functions different from those in the standard SVM. It constructs the following Lagrange function:

$$\begin{aligned} L_{\text{LS-SVM}} = & \frac{1}{2} \|\omega\|^2 + \frac{1}{2} \gamma \sum_{i=1}^l \xi_i^2 \\ & - \sum_{i=1}^l \alpha_i [y_i - f(x_i) - \xi_i], \end{aligned} \quad (6)$$

where α_i is the Lagrange multiplier. According to KKT optimal conditions, the linear equations can be obtained as follows:

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \Omega + \frac{I}{\gamma} \end{bmatrix} \begin{bmatrix} b^* \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \quad (7)$$

where $y = [y_1, y_2, \dots, y_l]^T$, $1_v = [1, 1, \dots, 1]^T$, $1_v = [1, 1, \dots, 1]^T$, I is the $l \times l$ identity matrix, Ω is a square matrix, and $\Omega_{i,j} = K(x_i, x_j)$ is the i th row and the j th column data element, where $\Omega_{i,j} = K(x_i, x_j)$ is the kernel function that satisfies the Mercer condition.

Solve the linear equations and get the nonlinear mapping equation as follows:

$$f(x) = \sum_{i=1}^l \alpha_i K(x_i, x) + b^*. \quad (8)$$

2.3. Weighted Extreme Learning Machine (WELM)

2.3.1. Basic Theory. In [6], the authors proposed a weighted extreme learning machine for imbalance learning, which defined an $N \times N$ diagonal matrix \mathbf{W} associated with every training sample x_i . Usually if training data x_i comes from a minority class (assumed to be positive class), the associated weight W_{ii} will be set relatively larger than others.

To maximize the marginal distance and to minimize the weighted cumulative error with respect to each sample, an optimization problem mathematically are written as

$$\text{Minimize: } \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|^2, \quad \|\boldsymbol{\beta}\|, \quad (9)$$

where $\mathbf{T} = [t_1, \dots, t_N]$. More precisely,

$$\text{Minimize: } L_{\text{ELM}} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}CW \sum_{i=1}^N \|\xi_i\|^2 \quad (10)$$

$$\text{Subject to: } \mathbf{h}(x_i)\boldsymbol{\beta} = t_i^T - \xi_i^T, \quad i = 1, \dots, N,$$

where $\mathbf{h}(x_i)$ is the feature mapping vector in the hidden layer with respect to x_i , $\boldsymbol{\beta}$ represents the output weight vector connecting the hidden layer and output layer, and C is the regularization parameter to represent the trade-off between the minimization of training errors and the maximization of the marginal distance. ξ_i , the training error of sample x_i , is caused by the difference of the desired output t_i and the actual output $\mathbf{h}(x_i)\boldsymbol{\beta}$.

2.3.2. Weighting Schemes. The key issue of the WELM is to define an appropriate weight matrix $\mathbf{W} = \text{diag}\{W_{ii}\}$, $i = 1, \dots, N$, which determines what degree of rebalance users are seeking for and how much further the boundary is pushed towards the majority class [6]. In [6], two weighting schemes are proposed.

The simple one is the weight value that can be automatically generated from the class information, which is in fact a special case of the cost sensitive learning:

$$\text{Weighting Scheme W1: } W_{ii} = \frac{1}{\#(t_i)}, \quad (11)$$

where $\#(t_i)$ is the number of samples belonging to class t_i , $i = 1, \dots, m$.

Another weighting scheme is the authors of [6] adopts the value of golden standard that represents the perfection in nature and finishes the balancing step into the ratio of 0.618 : 1 between minority classes and the majority classes, as shown in

Weighting Scheme W2

$$\begin{cases} W_{ii} = \frac{0.618}{\#(t_i)} & \text{if } t_i > \text{AVG}(t_i) \\ W_{ii} = \frac{1}{\#(t_i)} & \text{if } t_i \leq \text{AVG}(t_i). \end{cases} \quad (12)$$

Compared to weighting scheme W1, the boundary using weighting scheme W2 is pushed slightly backwards the minority class so that the misclassification cases in compromise on the majority side are sought of being alleviated, so we adopt weighting scheme W2 in the next experiments.

3. Modified Extreme Learning Machine Algorithm

The traditional extreme learning machines are based on the empirical risk minimization principle and the training error

minimization principle, whose drawback is that it is likely to suffer from overfitting, which reduces the generalization capability consequently.

According to the statistical theory, the actual risks include the empirical and structural risks, and a model with good generalization performance should be able to balance empirical and structural risks to obtain the best compromise. So we lead the structural risk minimization principle into the ELM algorithm and propose a modified weighted ELM and WELM model based on ELM and WELM, which we call it as M-ELM and M-WELM.

Assume that an input and output sample data set for regression analysis is $T = \{(x_1, y_1), \dots, (x_l, y_l)\}$, where $x_i \in R^n$ and $y \in R$, $i = 1, \dots, l$. We draw into the condition of the structural risk and adjust the proportion of the empirical and structural risks by ζ instead of the C in formula (10), and the optimization model of the optimal regression function can be established as follows:

$$\min \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}\zeta \sum_{i=1}^N \|\delta_i\|^2 \quad \text{for M-ELM}$$

or

$$\min \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}\zeta W \sum_{i=1}^N \|\delta_i\|^2 \quad \text{for M-WELM} \quad (13)$$

$$\text{Subject to } \begin{cases} y_i - \boldsymbol{\beta}_i \mathbf{h}(x_i) = \delta_i \\ \delta_i \geq 0, \quad i = 1, \dots, N, \end{cases}$$

where δ_i^2 , the sum of the square errors, represents the empirical risk and $\|\boldsymbol{\beta}\|^2$ represents the structural risk, according to the maximal margin principle in statistical theory [2]. According to formula (6), the formula above is the conditional extreme problem and can be transformed into the Lagrange equation as follows:

$$\begin{aligned} L_{\text{ELM}} = & \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}\zeta W \sum_{i=1}^N \|\delta_i\|^2 \\ & - \sum_{i=1}^N \alpha_i [y_i - \boldsymbol{\beta} \mathbf{h}(x_i) - \delta_i], \end{aligned} \quad (14)$$

where the Lagrange multiplier α_i is the constant factor of sample x_i in the linear combination to form the final decision function. Further, by making the partial derivatives with respect to variables $(\boldsymbol{\beta}, \delta_i, \alpha_i)$ all equal to zero, the KKT optimality conditions are obtained:

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = 0 \implies \boldsymbol{\beta} = \sum_{i=1}^N \alpha_i \mathbf{h}(x_i) = H^T \boldsymbol{\alpha}, \quad (15)$$

$$\frac{\partial L}{\partial \xi_i} = 0 \implies \alpha_i = \zeta W \delta_i, \quad i = 1, \dots, N, \quad (16)$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \implies y_i - \boldsymbol{\beta}_i \mathbf{h}(x_i) - \delta_i = 0. \quad (17)$$

The solution of $\boldsymbol{\beta}$ can be derived from (17) regarding left pseudoinverse. Usually, left pseudoinverse is more suitable

since it is much easier to compute matrix inversion of size $L \times L$, when L is much smaller than N :

$$\beta = \left(\frac{1}{\zeta} + H^T W H \right)^{-1} H^T W T. \quad (18)$$

The same as formula (7) in Section 2.2, we can obtain the following linear equations:

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \Omega + \frac{I}{\zeta} \end{bmatrix} \begin{bmatrix} 0 \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \quad (19)$$

where $y = [y_1, y_2, \dots, y_N]^T$; $1_v = [1, 1, \dots, 1]^T$; $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_N]$; I is the $N \times N$ identity matrix; Ω is a square matrix, and the i th row and the j th column data element is

$$\begin{aligned} \Omega_{i,j} &= K_{\text{ELM}}(x_i, x_j) = \mathbf{h}(x_i) \cdot \mathbf{h}(x_j) \\ &= [G(a_1, b_1, x_i), \dots, G(a_N, b_N, x_i)] \\ &\quad \cdot [G(a_1, b_1, x_j), \dots, G(a_N, b_N, x_j)]^T, \end{aligned} \quad (20)$$

where G is the excitation function.

The sigmoid function is used in this paper as follows:

$$G(a, b, x) = \frac{1}{(1 + \exp(-(a \cdot x) + b))}. \quad (21)$$

Solve the linear equations and then get the following nonlinear mapping equation below that is derived from (8):

$$f(x) = \sum_{i=1}^N \alpha_i K_{\text{ELM}}(x_i, x). \quad (22)$$

The whole steps of the M-ELM or M-WELM algorithm can be summarized as follows.

Given a training set T , activation function G , and hidden node number N , consider the following.

Step 1. Transform formula (13) of conditional extreme problem into formula (9) of Lagrange equation.

Step 2. Calculate α_i using formulas (16) and (17).

Step 3. Substitute α_i into formula (15) and calculate the output weight β .

As it can be seen, the M-WELM is able to be generalized to cost sensitive learning and can also deal with data with imbalanced class distribution as the WELM. On the other hand, its overfitting risk can be reduced by considering both the empirical and structural risks simultaneously.

4. Experiments and Performance Assessment

In this section, we present the performance comparison of proposed M-WELM and M-ELM and current reported ELM, OS-ELM [2], EOS-ELM [3], B-ELM [15], and C-ELM [14], and classifiers on benchmark prediction data sets first and then show the results for the image quality estimation problem. In the next whole experiments, the activation functions of ELM adopt sigmoid functions, and other parameters are gained by the cross validation method.

TABLE 1: The experimental results of the several algorithms on the Boston Housing data set.

Algorithm	Training error	Prediction error
M-WELM	2.253	3.364
M-ELM	2.154	3.592
W-ELM [6]	2.391	3.624
ELM	2.723	4.090
OS-ELM [2]	2.538	3.912
EOS-ELM [3]	2.686	3.745
B-ELM [12]	2.930	3.999
C-ELM [13]	2.615	3.653

4.1. Test on the Benchmark Boston Housing Data Set. Boston Housing data, obtained from the UCI database, is a data set commonly used for measuring the performance of regression algorithm. It contains the information of 506 sets of commodity houses in Boston Housing, including 12 continuous characteristics, one discrete characteristic, and house prices [16]. The purpose of regression estimation is to predict the average house price by training part of the samples.

In the experiments, the samples are randomly divided into two sample groups: random 70% of them for training and the remaining 30% for test. We repeat the random train-test procedure 100 times and calculate the mean square training and prediction error of every algorithm, and the experimental results of several algorithms are shown in Table 1. We have adjusted the parameters for every algorithm so that every algorithm can get a pretty good result. The number of hidden neurons of M-WELM, M-ELM, W-ELM, OS-ELM, and EOS-ELM is set to 180, and the number of hidden neurons of ELM, B-ELM, and C-ELM is set to 65.

It can be seen from Table 1, for the Boston Housing data set from a real-world multi-input single-output system, that our proposed M-WELM algorithm shows the best prediction performance than other types of ELMs, and M-ELM ranks number 2, with both of which indicating the robustness of our proposed idea for modifying ELM algorithm.

4.2. Test on the LIVE IQA Database. Algorithms that automatically assess perceptual image quality are critical for numerous image processing applications. Recently, machine learning based blind image quality assessment has great progress, such as the BRISQUE [17], the LBIQ [18], the DIIVINE [19], and the BLIINDS [20] using SVR and the GRNN-based method [21]. Of these indices, the BRISQUE shows the best performance in overall, so here we use the same image features adopted by the BRISQUE index [17] to test our proposed MELM and M-WELM algorithms for image quality assessment (IQA).

In reference [17], the authors used 36 natural scene statistical features in the spatial domain to predict image quality as shown in Table 2, that is, the shape and variance from a GGD fit of the MSCN coefficients, the shape, mean, left variance, and right variance from a GGD fit of the H pairwise products, V pairwise products, D1 pairwise products, and D2 pairwise products, which are extracted at two scales, the

TABLE 2: Summary of natural scene statistical features extracted in the spatial domain [14].

Feature ID	Feature description	Computation procedure
f_1-f_2	Shape and variance	Fit GGD to MSCN coefficients
f_3-f_6	Shape, mean, left variance, and right variance	Fit AGGD to H pairwise products
f_7-f_{10}	Shape, mean, left variance, and right variance	Fit AGGD to V pairwise products
$f_{11}-f_{14}$	Shape, mean, left variance, and right variance	Fit AGGD to D1 pairwise products
$f_{15}-f_{18}$	Shape, mean, left variance, and right variance	Fit AGGD to D2 pairwise products

TABLE 3: Median Spearman's rank ordered correlation coefficient (SROCC), RMS, and running time across 1000 train-test combinations on the LIVE IQA database (80% samples are used for training).

Algorithm	SROCC (%)	RMS	Running time (s)
M-WELM	94.055	9.349	38.33
W-ELM [6]	93.325	9.566	37.60
M-ELM	93.312	9.451	30.17
ELM	93.115	10.240	46.96
OS-ELM [2]	93.159	9.973	227.44
EOS-ELM [3]	93.207	10.189	173.14
B-ELM [12]	91.288	10.541	343.41
C-ELM [13]	92.508	11.109	150.83
SVR	93.95	9.767	1763.97

original image scale, and at a reduced resolution (low pass filtered and downsampled by a factor of 2). Here we also adopt these 36 image statistical features to predict image quality by using different ELM algorithms and then compare our proposed modified ELM algorithms with the reported ELM algorithms.

Firstly, we test out proposed algorithm on the LIVE IQA database [22], which consists of 29 reference images with 779 distorted images spanning five different distortion categories" JPEG2000 (JP2K) and JPEG compression, additive white Gaussian noise (WN), Gaussian blur (Blur), and a Rayleigh fast-fading channel simulation (FF). Each of the distorted images has an associated difference mean opinion score (DMOS) which represents the subjective quality of the image.

Three performance metrics are used to evaluate the algorithms. The first is the Spearman rank ordered correlation coefficient (SROCC), which measures the prediction monotonicity of the quality index. The second is the root mean square error (RMSE). The third is the running time.

Because learning based method requires a training stage in order to construct the relationship between the extracted statistical features and DMOS, we split the LIVE dataset into two nonoverlapping sets—a training set and a testing set. The training set consists of 80%, 50%, or 30% of the 29 reference images and their associated distorted versions, respectively, while the testing set consists of the remaining 20%, 50%, or 70% of the 29 reference images and their associated distorted versions. The regression models are trained on the training set and the results are then tested on the testing set. In order to ensure that the proposed method is robust across content and is not governed by the specific train-test split utilized, we repeat this random 80% train—20% test,

TABLE 4: Median Spearman's rank ordered correlation coefficient (SROCC), RMS, and running time across 1000 train-test combinations on the LIVE IQA database (50% samples are used for training).

Algorithm	SROCC (%)	RMS	Running time (s)
M-WELM	92.319	10.471	51.15
W-ELM [6]	91.931	10.563	45.45
MELM	92.140	10.494	42.54
ELM	90.170	11.391	56.69
OS-ELM [2]	90.920	11.260	120.25
EOS-ELM [3]	91.017	11.231	119.55
B-ELM [12]	90.324	11.384	112.01
C-ELM [13]	88.251	12.117	115.05
SVR	90.263	11.796	857.91

TABLE 5: Median Spearman's rank ordered correlation coefficient (SROCC), RMS, and running time across 1000 train-test combinations on the LIVE IQA database (30% samples are used for training).

Algorithm	SROCC (%)	RMS	Running time (s)
M-WELM	90.340	11.621	63.75
W-ELM [6]	89.891	11.712	53.70
MELM	90.069	11.700	54.38
ELM	87.859	12.966	49.87
OS-ELM [2]	88.180	12.885	265.61
EOS-ELM [3]	88.304	12.853	271.91
B-ELM [12]	88.156	13.179	62.48
C-ELM [13]	86.316	14.061	72.40
SVR	87.655	13.110	415.63

50% train—50% test, and 30% train—70% test split 1000 times on the LIVE dataset and evaluate the performance on each of these test sets. The median Spearman rank ordered correlation coefficient (SROCC), RMS, and running time across these 1000 train-test trials are reported in Tables 3, 4, and 5. For comparison, the results of SVR and other current reported ELM algorithms are also listed in Tables 3–5, using the same random train-test procedure 1000 times. The SVR is implemented by utilizing the libSVM package [23]. The kernel used for SVR is the radial basis function (RBF) kernel, whose parameters are estimated using cross validation on the training set. Other ELM algorithms are implemented by us. The used number of hidden neurons of M-WELM, M-ELM, W-ELM, OS-ELM, and EOS-ELM is set to 120, and the number of hidden neurons of ELM, B-ELM, and C-ELM is set to 75.

TABLE 6: Median Spearman's rank ordered correlation coefficient (SROCC), RMS, and running time across 1000 train-test combinations on the LIVE IQA and TID 2008 database.

Algorithm	SROCC (%)	RMS	Running time (s)
M-WELM	89.012	0.713	35.01
W-ELM [6]	88.762	0.72	28.94
MELM	88.879	0.718	21.85
ELM	87.756	0.752	28.78
OS-ELM [2]	87.630	0.758	102.38
EOS-ELM [3]	87.720	0.758	102.83
B-ELM [12]	87.750	0.752	62.66
C-ELM [13]	86.761	0.758	112.58
SVR	89.637	0.702	213.72

As it can be seen from Tables 3–5, compared with ELM (or WELM), our proposed M-ELM (or M-WELM) shows better subjective judgment no matter how much the percentage of samples is used, which suggests the validity of introducing the structural risk minimization principle. Our M-WELM algorithm shows the best performance against the other reported ELM algorithm and SVR algorithm, especially when using less training samples, which further demonstrates the effectiveness of integrating the structural risk minimization principle and the weight method into the ELM model. In addition, we can find that our proposed M-WELM is far faster than the SVR, which provides an effective real-time solution to IQA.

4.3. Test on the TID 2008 Database. To prove the promotion of the proposed M-WELM, we further test on the same (available) distortions in an alternate database—the TID2008 [24]. It consists of 25 reference images and 17 distortion types with 1700 distorted images. Of these 25 reference images only 24 are natural images, so we test our algorithm only on these 24 images. Here we use all 779 distorted images in the LIVE IQA database as the training set and the images in the TID 2008 as the testing set. We still repeat the random train-test procedure 1000 times and report the median SROCC, RMS, and running time as shown in Table 6. The values of parameters of every algorithm are the same as the used in Section 4.2.

From Table 6, we can find our proposed M-WELM showing the highest consistency with the subjective scores amongst all types of ELM algorithms, and it is also competitive with the SVR in the performance, but it is far faster than the SVR, which provides a real-time solution to IQA.

5. Conclusion

Current reported ELM and weighted ELM algorithms are based on empirical risk minimization principle, which may easily lead to the overfitting risk during learning process. By introducing the structural risk minimization principle to the ELM and weighted ELM algorithms, we propose an improved (weighted) extreme learning machine algorithm (M-WELM and M-ELM) to solve the overfitting problem, which takes into account both the empirical risk and the structural risk simultaneously and adjusts the proportion of the two risks

properly. Our experimental results show that the M-WELM outperforms the current reported ELM algorithms in IQA and also has competitive performance with the SVR, but it is far faster than the SVR, which provides an effective real-time solution to IQA.

Conflict of Interests

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

Acknowledgments

This research is supported in part by the National Natural Science Foundation of China (no. 61170120), Program for New Century Excellent Talents in University (NCET-12-0881), the Natural Science Foundation of Jiangsu Province (no. BK2011147), China Agriculture Research System (CARS-49), and the Fundamental Research Funds for the Central Universities (JUSRP51410B).

References

- [1] 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.
- [2] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.
- [3] Y. Lan, Y. C. Soh, and G.-B. Huang, "Ensemble of online sequential extreme learning machine," *Neurocomputing*, vol. 72, no. 13–15, pp. 3391–3395, 2009.
- [4] H.-J. Rong, G.-B. Huang, N. Sundararajan, and P. Saratchandran, "Online sequential fuzzy extreme learning machine for function approximation classification problems," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 39, no. 4, pp. 1067–1072, 2009.
- [5] G. Feng, G.-B. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Transactions on Neural Networks*, vol. 20, no. 8, pp. 1352–1357, 2009.
- [6] W. Zong, G.-B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, 2013.
- [7] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.
- [8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [9] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.
- [10] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing*, vol. 48, no. 1–4, pp. 85–105, 2002.
- [11] J. A. K. Suykens and J. Vandewalle, "Training multilayer perceptron classifiers based on a modified support vector method," *IEEE Transactions on Neural Networks*, vol. 10, no. 4, pp. 907–911, 1999.

- [12] Y. Bian, J. Yang, M. Li, and R. Lan, "Automated flare prediction using extreme learning machine," *Mathematical Problems in Engineering*, vol. 2013, Article ID 917139, 7 pages, 2013.
- [13] C.-J. Lu and Y. E. Shao, "Forecasting computer products sales by integrating ensemble empirical mode decomposition and extreme learning machine," *Mathematical Problems in Engineering*, vol. 2012, Article ID 831201, 15 pages, 2012.
- [14] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, no. 1-4, pp. 306-314, 2005.
- [15] Y. Yang, Y. Wang, and X. Yuan, "Bidirectional extreme learning machine for regression problem and its learning effectiveness," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1498-1505, 2012.
- [16] Y. Jian, W. Jue, and Z. Ning, "Laplacian semi-supervised regression on a manifold," *Journal of Computer Research and Development*, vol. 44, no. 7, pp. 1121-1127, 2007.
- [17] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695-4708, 2012.
- [18] H. Tang, N. Joshi, and A. Kapoor, "Learning a blind measure of perceptual image quality," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 305-312, June 2011.
- [19] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: from natural scene statistics to perceptual quality," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3350-3364, 2011.
- [20] M. A. Saad, A. C. Bovik, and C. Charrier, "Blind image quality assessment: a natural scene statistics approach in the DCT domain," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3339-3352, 2012.
- [21] C. Li, A. C. Bovik, and X. Wu, "Blind image quality assessment using a general regression neural network," *IEEE Transactions on Neural Networks*, vol. 22, no. 5, pp. 793-799, 2011.
- [22] H. R. Sheikh, Z. Wang, L. K. Cormack, and A. C. Bovik, LIVE Image Quality Assessment Database, <http://live.ece.utexas.edu/research/quality>.
- [23] C. Chang and C. Lin, LIBSVM: A Library for Support Vector Machines, 2001, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
- [24] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "Tid2008—a database for evaluation of full reference visual quality assessment metrics," *Advances of Modern Radioelectronics*, vol. 10, pp. 30-45, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

