Article

A Novel Orthogonal Extreme Learning Machine for Regression and Classification Problems

Licheng Cui ^{1,2,*} , Huawei Zhai ^{3,*} and Hongfei Lin ¹

¹ Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China; hflin@dlut.edu.cn

² Public Security Information Department, Liaoning Police College, Dalian 116036, China

³ Information Science and Technology School, Dalian Maritime University, Dalian 116026, China

* Correspondence: clcLicheng@163.com (L.C.); zhw@dlmu.edu.cn (H.Z.)

Received: 18 August 2019; Accepted: 10 October 2019; Published: 14 October 2019

Abstract: An extreme learning machine (ELM) is an innovative algorithm for the single hidden layer feed-forward neural networks and, essentially, only exists to find the optimal output weight so as to minimize output error based on the least squares regression from the hidden layer to the output layer. With a focus on the output weight, we introduce the orthogonal constraint into the output weight matrix, and propose a novel orthogonal extreme learning machine (NOELM) based on the idea of optimization column by column whose main characteristic is that the optimization of complex output weight matrix is decomposed into optimizing the single column vector of the matrix. The complex orthogonal procrustes problem is transformed into simple least squares regression with an orthogonal constraint, which can preserve more information from ELM feature space to output subspace, these make NOELM more regression analysis and discrimination ability. Experiments show that NOELM has better performance in training time, testing time and accuracy than ELM and OELM.

Keywords: extreme learning machine; orthogonal constraint; orthogonal procrustes problem; least squares regression

1. Introduction

An extreme learning machine (ELM) is an innovative learning algorithm for the single hidden layer feed-forward neural networks (SLFNs for short), proposed by Huang et al [1], that is characterized by the internal parameters generated randomly without tuning. In essence, the ELM is a special artificial neural network model, whose input weights are generated randomly and fixed, so as to get the unique least-squares solution of the output weight [1], making the performance better [2–4]. In the conventional model there is a lack of convergence ability, generalization, over-fitting, local minimum and parameter adjustment, all of which make the ELM superior [1,5]. Considering the learning process of the ELM, it is relatively simple. Firstly, some internal parameters of the hidden layer are generated randomly, such as input weights connecting the input layer and hidden layer, the number of hidden layer neurons, etc., which are fixed during the whole process. Secondly, the non-linear mapping function is selected to map the inputting data to the feature space, and through analyzing the real output results and expected output results, the key parameter (output weight connecting the hidden layer and the output layer) can be directly obtained, omitting iterative tuning. So, its training speed is considerably faster than that of the conventional algorithms [6].

Due to the good performance of the ELM, it is used widely in regression and classification. For meeting higher requirements, researchers optimize and improve the ELM, and have proposed many better algorithms based on ELM. Di Wang et al. combined the local-weighted jackknife and RELM, proposed a novel conformal regressor (LW-JP-RELM), which complements ELM with interval

predictions satisfying a given level of confidence [7]. For improving the generalization performance, Ying Yin et al. proposed enhancing the ELM by a Markov boundary-based feature selection, based on the feature interaction and the mutual information to reduce the number of features, so as to construct more compact network, whose generalization was improved greatly [8]. Ding et al. reformulated an optimization extreme learning machine to take a new regularization parameter, which is bounded between 0 and 1, and also easier to interpret as compared to the error penalty parameter C , it could achieve better generalization performance [9]. For solving sensibility of ELM to the ill-conditioned data, Hasan et al. proposed two novel algorithms based on ELM, ridge regression and almost unbiased ridge regression, and also gave three criteria to select the regularization parameter, which improved the generalization and stability of the ELM greatly [10]. Besides, there are more effective algorithms based on the ELM, such as Distributed Generalized Regularized ELM (DGR-ELM) [11], Self-Organizing Map ELM (SOM-ELM) [12], Data and Model Parallel ELM (DMP-ELM) [13], Genetic Algorithm ELM (GA-ELM) [14], Jaya optimization with mutation ELM (MJaya-ELM) [15], et al.

Either with a simple ELM or more complex algorithms based on ELM one must find the optimal solution of the two key parameters in ELM essentially, the number of hidden layer neurons and the output weights. From input layer to the output layer, essentially ELM learns the output weights based on the least squares regression analysis [16]. Therefore, many algorithms except those mentioned above are still proposed based on least squares regression, and their main work is to find an optimal transformation matrix, so as to minimize the error of sum-of-squares. Among these strategies [17,18], introducing orthogonal constraint into the optimization problem is required and also employed widely in the classification and subspace learning. Nie et al. showed that the performance of least squares discriminant and regression analysis after introducing orthogonal constraint is much better than those without orthogonal constraint [19,20]. After introducing orthogonal constraint into ELM, the optimization problem is seen as unbalanced procrustes problems, which is hard to be solved. Yong Peng et al. pointed out that the unbalanced procrustes problem can be transformed into a balanced procrustes problem, which is relatively simple [16]. Motivated by this research, in this paper we focus on the output weight, a novel orthogonal optimizing method (NOELM) is proposed to solve the unbalanced procrustes problem, and its main contribution is that the optimization of complex matrix is decomposed into optimizing the single column vector of the matrix, reducing the complexity of the algorithm.

The remainder of the paper is organized as follows. Section 2 reviews briefly the basic ELM model. In Section 3, the model formulation and the iterative optimization method are detailed. The convergence and complexity analysis is presented in Section 4. In Section 5, the experiments are conducted to show the performances of NOELM. Finally, Section 6 concludes the paper.

2. Extreme Learning Machine

Mathematically, given N discrete sample $(x_i, y_i)_{i=1}^N$, where N is the sample number, $x_i \in R^n$ is the input vector, $y_i \in R^m$ is the expected output vector, and the expected output of the i -th sample is y_i , $y_i = [y_{i1} \ y_{i2} \ \dots \ y_{im}]^T$, $i = 1, 2, 3, \dots, N$. For selected activation function, if the real output of the SLFNs is the same as the expected output y_i , the mathematical representation of SLFNs is as follows:

$$f(x_j) = \sum_{i=1}^L \beta_i g(\omega_i, x_j, b_i) = y_j, \quad (1)$$

where $\omega_i = [\omega_{i1}, \dots, \omega_{in}]^T$ is the input weight connecting the input layer to the i -th hidden layer neuron, b_i is the basis of the i -th hidden layer neuron, $\beta_i = [\beta_{i1}, \dots, \beta_{im}]^T$ is the output weight connecting the i -th hidden layer neuron and the output layer, and L is the number of the hidden layer neurons, shown in Figure 1.

Figure 1. The Architecture of ELM Model.

Equation (1) can be compactly rewritten as

$$H\beta = Y, \quad (2)$$

where

$$H = \begin{bmatrix} g(\omega_1, x_1, b_1) & \cdots & g(\omega_L, x_1, b_L) \\ \vdots & \ddots & \vdots \\ g(\omega_1, x_N, b_1) & \cdots & g(\omega_L, x_N, b_L) \end{bmatrix}_{N \times L}, \quad (3)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \quad Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times m}. \quad (4)$$

So, based on the theory of ELM, the optimal solution of Equation (2) is as follows,

$$\beta = H^+ Y, \quad (5)$$

where H^+ is the Moore–Penrose inverse of the Matrix H , $H^+ = (H^T H)^{-1} H^T$. For further improving model precision, the regularization is introduced into ELM, the optimal problem is transformed as follows,

$$\min : \frac{1}{2} \|H\beta - Y\|^2 + \frac{C}{2} \|\beta\|^2, \quad (6)$$

where C is the regularization parameter, which is used to balance the empirical risk and structural risk. Based on the Karush–Kuhn–Tucker condition, the optimal solution of β is obtained:

$$\beta = (H^T H + CI)^{-1} H^T Y. \quad (7)$$

3. Novel Orthogonal Extreme Learning Machine (NOELM)

The orthogonal constraint is introduced into ELM, shown in Figure 1, the optimal problem is transformed as follows,

$$J(\beta_{L+1}) = \min_{\beta_{L+1}^T \beta_{L+1} = I} \|H_{L+1} \beta_{L+1} - Y\|^2, \quad (8)$$

where $H_{L+1} \in R^{N \times (L+1)}$, $\beta_{L+1} \in R^{(L+1) \times m}$ is the output matrix and the output weight of the hidden layer, $Y \in R^{N \times m}$. Because of the orthogonal constraint, the input samples are mapped into an orthogonal subspace, where their metric structure could be preserved.

Set $L > m$, so the problem (8) is an unbalanced orthogonal procrustes problem which is difficult to be resolved directly because of the orthogonal constraint [16]. In this paper, an improved method is proposed to optimize the problem (8) based on the following lemma.

Lemma 1 [[21], Theorem 3.1]. If $\beta_{L+1}^* = [\rho_1^*, \dots, \rho_{L+1}^*]$ is the optimal solution for the problem (8) and its orthogonal complement is B_{L+1}^* , then $H_{L+1}^T [Y, H_{L+1}^T B_{L+1}^*]$ is positive, semi-definite and symmetric, and

$$\|H_{L+1}^T \rho_j^* - y_j\| = \min_{\rho_j \perp \beta_j^*, \|\rho_j\|^2=1} \|H_{L+1} \rho_j - \hat{y}_j\|, \quad (9)$$

where $\tilde{\beta}_j^* = [\rho_1^*, \dots, \rho_{j-1}^*, \rho_{j+1}^*, \dots, \rho_{L+1}^*]$, $\hat{y}_j$ is the j -th column vector of Y .

The proof of Lemma 1 is simple, which can be found in the literature [21]. Motivated by Lemma 1, a local transformation is applied in the Equation (8), we relax the j -th column ρ_j ($j \leq (L+1)$) and fix others, $\tilde{\beta}_j = [\rho_1, \dots, \rho_{j-1}, \rho_{j+1}, \dots, \rho_{L+1}]$, then the equation could be transformed into

$$J(\rho_j) = \min_{\rho_j \perp \tilde{\beta}_j, \|\rho_j\|^2=1} \|H_{L+1} \rho_j - \hat{y}_j\|^2. \quad (10)$$

If ρ_j^* is the optimal solution of Equation (10), the approximation β_{L+1} could be improved after replacing ρ_j by ρ_j^* , and obviously, the modified $\beta_{L+1}^* = [\rho_1, \dots, \rho_{j-1}, \rho_j^*, \rho_{j+1}, \dots, \rho_{L+1}]$ is also orthogonal.

To resolve the constrained problem (10) is a little difficult, so, the orthogonal complement B_{L+1} of β_{L+1} can be used to simplify the Equation (10). Set $P_{L+1} = [\rho_j, B_{L+1}]$, and it is known $\rho_j \perp \tilde{\beta}_j$, then P_{L+1} is the orthogonal complement of $\tilde{\beta}_j$. So, in the constrained problem (10), the condition $\rho_j \perp \tilde{\beta}_j$ could be represented in another form, $\rho_j = P_{L+1}x$, $x \in R^{n-L}$ is a unit vector. Thus, the problem (10) can be transformed into the following form with quadratic equality constraint:

$$J(x) = \min_{x^2=1} \|H_{L+1} P_{L+1} x - \hat{y}_j\|^2. \quad (11)$$

Clearly, after get the optimal solution x^* of problem (11), the solution of problem (10) is $\rho_j = P_{L+1}x^*$. If the orthogonal complement of β_{L+1}^* is B_{L+1}^* , then $B_{L+1}^* = P_{L+1}W$, W is the orthogonal complement of x^* , and it can be constructed easily, using the Householder reflection $I - 2u\omega^T$ with $\|\omega\| = 1$, which meets $(I - 2\omega\omega^T)x^* = -\text{sign}(x_1^*)e_1$, x_1^* is the first component of x^* . Indeed, partitioning $P_{L+1}(I - 2\omega\omega^T)$, ρ_j^* and B_{L+1}^* can be picked out from the following equation,

$$P_{L+1}(I - 2\omega\omega^T) = [-\text{sign}(x_1^*)\rho_j^*, B_{L+1}^*]. \quad (12)$$

For resolving the problem (11), it first rewrites the Equation (11) in general form,

$$J(x) = \min_{\|x\|^2=1} \|Ax - y\|^2, \quad (13)$$

where $A = H_{L+1}P_{L+1}$, $y = \hat{y}_j$. $\|Ax - y\|^2$ is transformed in the following form

$$\begin{aligned} \|Ax - y\|^2 &= \|Ax\|^2 + \|y\|^2 - 2\text{trace}(x^T A^T y) \\ &\leq \|A\|^2 + \|y\|^2 - 2\text{trace}(x^T A^T y). \end{aligned} \quad (14)$$

Known from the Equations (13) and (14), the parameters A and y are fixed, the minimum problem of function $J(x)$ is transformed to the maximum of $\text{trace}(x^T A^T y)$ approximately, showing by Equation (15), denoted by $W = A^T y$.

$$x = \{X : X = \text{argmax trace}(X^T W)\}. \quad (15)$$

Let Singular Value Decomposition of W be

$$W = U \text{diag}(\Sigma_k, O_{s-k}) V^T, \quad (16)$$

where $x \in R^{n-L}$, $s = n - L$, $\Sigma_k = \text{diag}(\sigma_1, \dots, \sigma_k)$, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k > 0$, $k = \text{rank}(W)$, U and V are orthogonal.

Set $X^* = U^T X V$,

$$\text{trace}(X^T W) = \text{trace}(X^* \text{diag}(\Sigma_k, O_{s-k})). \quad (17)$$

So,

$$x = \left\{ X : X = U X^* V^T, X = \arg \max \text{trace}(X^* \text{diag}(\Sigma_k, O_{s-k})) \right\}, \quad (18)$$

As known above, $x \in R^{n-L}$ is a unit vector, partitioning X^*

$$X^* = \begin{bmatrix} X_{11} \\ X_{21} \end{bmatrix}, X_{11} \in R^k, X_{21} \in R^{(s-k)}. \quad (19)$$

Because of $x \in R^{n-L}$, then

$$\max \text{trace}(X^* \text{diag}(\Sigma_k, O_{s-k})) = \max \text{trace}(X_{11}^T \Sigma_k). \quad (20)$$

Note that $X^* = [x_{ij}]$ is unit and orthogonal, $\|X^*\| = 1$, then $-1 \leq x_{ij} \leq 1$, so based on the Equation (20), for the maximum, it can be deduced that $x_{ij} = 1$, $i = j$, then $X_{11} = I_k$, $X_{21} \in O^{s-k}$, and $k = 1$. Hence, $G \in O^{s-k}$,

$$x = \left\{ X : X = U \begin{bmatrix} I_k \\ G \end{bmatrix} V^T \right\}. \quad (21)$$

Based on the analysis above, the novel optimization to objective problem (8) is proposed in the position, its detail is as follows (Algorithm 1):

Algorithm 1: Optimization to objective problem (8)

Basic Information: training samples $\{(x_i, y_i)_{i=1}^N \mid x_i \in R^n, y_i \in R^m\}$

Initialization: Set threshold τ and η

- S1. Generate the input weight layer w and bias vector b ;
 - S2. Calculate the output matrix of the hidden layer H based on Equation (3);
 - S3. Calculate the orthogonal β' of $\text{span } H^T Y$, and its orthogonal complement B_{L+1} , then $r_0 = \|H\beta' - Y\|$;
 - S3. **While** $j = 1, 2, \dots, m$
 - S4. Relax the j -th column $\rho_j, \hat{y}_j$ from the matrix β, Y separately, and fix the rest;
 - S5. Set $P = [\rho_j, B_{L+1}]$, then solve $x = \arg \min_{\|x\|=1} \|HPx - \hat{y}_j\|^2$;
 - S6. Set $A = HP$, $y = \hat{y}_j$, then $W = A^T y$. By SVD, $W = U \text{diag}(\Sigma_k, O_{s-k}) V^T$, so as to obtain U and V ;
 - S7. Based on the Equation (21), $x = U I_k V^T$;
 - S8. Calculate the vector $u = (x + se_1) / \|x + se_1\|$, $s = \text{sign}(x)$;
 - S9. Partition $P(I - 2uu^T) = [-s\rho_j^*, B_{L+1}^*]$ so as to obtain ρ_j^* and B_{L+1} , then replace ρ_j of β by ρ_j^* to obtain $\beta^*, B_{L+1} = B_{L+1}^*$ and $\beta = \beta^*$;
 - End While**
 - S10. Calculate $r_1 = \|H\beta^* - Y\|$, then if $(r_0 - r_1) < \tau r_1$ $\|r_1\| \leq \eta$, terminate, otherwise, $r_0 = r_1$, B_{L+1} is the new orthogonal complement of β^* , go to step S3.
-

4. Convergence and Complexity Analysis

Considering the convergence if the algorithm, Let $\{\beta^i, j\}$ be a sequence of β^* generating during iteration, which converges to β , so, its orthogonal complement $B^{i,j}$ also converges to B , where i is

the iterating number, and i is the operation of relaxing the j -th column from original matrix, so, it follows $P^{i,j} = [\rho_j^{i,j}, B^{i,j}]$ converges to $P = [\rho_j, B]$.

Based on the equations above, it is known that ρ_j^* is the optimal solution of Equation (10). If $\|H\rho_j - \hat{y}_j\| - \|H\rho_j^* - \hat{y}_j\| = \eta$, then for i is large enough, $P^{i,j}$ and P meets

$$\|P^{i,j} - P\| = \left\| \begin{bmatrix} \rho_j^{i,j} \\ B^{i,j} \end{bmatrix} - \begin{bmatrix} \rho_j \\ B \end{bmatrix} \right\| < \frac{\eta}{4\|H\|_2}, \quad (22)$$

Set $\rho_j^* = Px$, $\|x\| = 1$, and $\hat{\rho}_j = P^{i,j}x$, then based on Equation (22), it has

$$\|\hat{\rho}_j - \rho_j^*\| = \|P^{i,j}x - Px\| \leq \|P^{i,j} - P\| \|x\| < \frac{\eta}{4\|H\|_2}, \quad (23)$$

Based on Equations (10) and (22), it has

$$\|\hat{\rho}_j - \rho_j^*\| = \|P^{i,j}x - Px\| \leq \|P^{i,j} - P\| \|x\| < \frac{\eta}{4\|H\|_2}, \quad (24)$$

Based on the Equations (23) and (24), it has

$$\begin{aligned} f(\beta^{i,j}) - f(\beta^{i+1,j}) &= \|H\beta^{i,j} - Y\| - \|H\beta^{i+1,j} - Y\| \\ &= \|H\rho_j^{i,j} - \hat{y}_j\| - \min_{\rho \perp \beta_j^{i+1,j}, \|\rho\|=1} \|H\rho - \hat{y}_j\| \\ &\geq \|H\rho_j^{i,j} - \hat{y}_j\| - \|H\hat{\rho}_j - \hat{y}_j\| \\ &\geq \|(H\rho_j - \hat{y}_j) + H(\rho_j^{i,j} - \rho_j)\| + \|(H\rho_j^* - \hat{y}_j) + H(\hat{\rho}_j - \rho_j^*)\| \\ &\geq \|(H\rho_j - \hat{y}_j)\| - \|H(\rho_j^{i,j} - \rho_j)\| - \|(H\rho_j^* - \hat{y}_j)\| - \|H(\hat{\rho}_j - \rho_j^*)\| \\ &\geq \|(H\rho_j - \hat{y}_j)\| - \|H(\rho_j^{i,j} - \rho_j)\| - \|H(\rho_j^* - \rho_j)\| - \|H(\hat{\rho}_j - \rho_j^*)\| \\ &\geq \eta - \frac{\eta}{4} - \frac{\eta}{4} \\ &\geq \frac{\eta}{2} \end{aligned}$$

So, $f(\beta^{i,j}) - f(\beta^{i+1,j}) \geq \frac{\eta}{2}$, based on the derivation of the inequality above, it can be deduced that $f(\beta^{1,j}) > f(\beta^{2,j}) > \dots > f(\beta^{n,j})$. By the same method and analysis, it also can be obtained that $f(\beta^{i,1}) > f(\beta^{i,2}) > \dots > f(\beta^{i,m})$. So, it is $f(\beta^{1,1}) > f(\beta^{1,2}) > \dots > f(\beta^{2,1}) > \dots > f(\beta^{3,1}) > \dots$, so the sequence $\{f(\beta^{i,j})\}$ is monotonically decreasing, and when $i \rightarrow \infty$, $f(\beta^{i,j}) - f(\beta^{i+1,j}) \rightarrow 0$. In a word, after analysis above, the novel algorithm monotonically decreases the objective shown in Equation (8).

It is known that the complexity of ELM derives from the calculation of output weights β , or rather, it is mainly used to calculate the inverse of matrix $H^T H + CI$. In most cases, the number of hidden layer neurons L is much smaller than the training sample size N , $L \ll N$, thus the complexity is less than least square support vector machine (LS-SVM) and proximal support vector machine (PSVM), which need to calculate the inverse of $N \times N$ matrix [16]. As we know, the complexity of ELM and OELM is $O(L^3)$, $O(t(NL^2 + L^3))$ separately. As for the complexity of the novel algorithm proposed in the paper, its main calculation is from the loop. In each iteration, it needs to find the optimal solution of one column relaxing from β , and during this, it needs to do SVD decomposition on the $m \times 1$ matrix $A^T y$, whose complexity is $O(m^2)$, and then, the complexity of updating β once is $O(m)$. So, the complexity of the proposed algorithm is $O(tm^3)$, where t is the number of updating β . In real application, regardless of classification or regression, the output dimension is much less than the number of hidden layer neurons and the training samples size.

As we know, $(H\beta)^T = Y^T$, then $\beta^T h_i^T(x_i) = y_i$. Considering the Euclidean distance between any two data points y_i and y_j , because of the orthogonal constraint $\beta^T \beta = I$, it has $\|y_i - y_j\| = \|h_i(x_i) - h_j(x_j)\|$.

It is known that $h_i(x_i)$ is the point in the ELM feature space, $\|h_i(x_i) - h_j(x_j)\|$ is the distance in ELM space, and $\|y_i - y_j\|$ is the distance in the subspace. From this analysis, the novel ELM with orthogonal constraints is superior in maintaining the metric structure from first to last.

5. Performance Evaluation

For testing the performances of the novel algorithm proposed in the paper, it is compared with other learning algorithms on the classification problems (EMG for Gestures, Avila and Ultrasonic Flowmeter) and regression problems (Auto price, Breast cancer, Buston housing, etc.), which are from the University of California Irvine (UCI) machine learning repository [22], shown in Table 1. These learning algorithms include ELM [1], OELM [16] and I-ELM [23,24], their activation function is the sigmoid function, and the number of hidden layer neurons is set as three times as the input dimension. For I-ELM, the initial number of hidden layer neurons is set to zero. In the real experiments, the key parameters such as the input weights, the biases, etc., are generated randomly from $[-1, 1]$, and then, all samples are normalized into $[-1, 1]$, and the outputs of the regression problems are normalized into $[0, 1]$ [25]. All simulations are done in Matlab R2016a environment.

Table 1. The specification of the datasets.

Datasets	Training	Testing	Attributes	Class
Avila	5000	2000	10	12
Electro-Myo-Graphic data (EMG) for Gestures	10000	2000	6	8
Ultrasonic Flowmeter	112	69	33	4
Stock	450	500	9	-
Abalone	2000	1177	8	-
Auto price	80	79	14	-
Auto-Miles Per Gallon (MPG)	320	78	8	-
Breast cancer	100	94	32	-
Buston housing	250	256	13	-
California housing	8000	12640	8	-
Census house (8L)	10000	12784	8	-

In the classification problems, ELM and OELM are selected to compare with NOELM. The experimental results are shown in Figures 2 and 3. Figure 2 shows the convergence property of NOELM. At first, the convergence rate is larger, the objective value falls rapidly, when reaching about 0.8, it falls slowly, until stable. During the whole process, the number of iterations does not vary significantly, the maximum is not more than 20, and the minimum is only about 5, so in a word, the novel algorithm is a little more effective. Figure 3 shows the comparison of the training time and classification rate. Due to the complexity above, the traditional ELM is low in complexity, and its training time is shortest. The complexity of NOELM is less than OELM, then its training time is shorter than OELM, and longer than that of ELM because of too many iterations, but the difference is not larger than 0.05. Although NOELM is not the best in terms of training time, its classification is better than the other two, the largest rate can reach 0.9.

Figure 2. Convergence property of novel orthogonal ELM (NOELM).

Figure 3. Comparison of training time and classification rate of ELM, OELM and NOELM.

In the regression problems, ELM, OELM and I_ELM are selected to compare with NOELM, the experimental results are shown in Table 2. As mentioned above, the number of hidden layer neurons is determined based on the input dimension, so the hidden layer neurons of ELM, OELM and NOELM are fixed, and the others are dynamically increasing hidden layer neurons. Analyzing the information of Table 2, compared with I_ELM, the network complexity of NOELM is a little lower, and its structure is more compact, but it is a little worse than I_ELM in some datasets, the difference is not large and fully acceptable. As for the accuracy of training and testing from Tables 3 and 4, comparing with ELM and OELM, the performances of NOELM is better, it has better stability. Because of characteristics of I_ELM, it constructs a more compact network and is a little superior in the training and testing accuracy in some datasets, and this is just the weak point of NOELM and other related algorithms. However, by introducing the orthogonal constraints and improving the algorithm, NOELM can greatly narrow this gap, and its performance is also acceptable.

Table 2. Comparison of the network complexity and training time.

	ELM		OELM		I_ELM		NOELM	
	Nodes	Time(s)	Nodes	Time(s)	Nodes	Time(s)	Nodes	Time(s)
Auto price	42	0.0325	42	0.0677	50	0.0374	42	0.0241
Breast cancer	96	1.0217	96	2.1285	66	0.2324	96	0.7568
Buston housing	39	0.0453	39	0.0944	100	0.5672	39	0.0336
Auto-MPG	24	0.8835	24	1.8406	76	0.8173	24	0.6544
Stock	27	0.6392	27	1.3317	97	0.8039	27	0.4735
Abalone	24	0.4836	24	1.0075	40	0.3237	24	0.3582
California housing	24	0.4547	24	0.9473	69	6.0856	24	0.3368
Census house (8L)	24	0.7667	24	1.5973	57	5.2479	24	0.5679

Table 3. Comparison of the average of training and testing (Root Mean Square Error).

	ELM		OELM		I_ELM		NOELM	
	Train	Test	Train	Test	Train	Test	Train	Test
Auto price	0.1283	0.1297	0.1141	0.1212	0.0997	0.1089	0.1056	0.1161
Breast cancer	0.13182	0.1499	0.1163	0.1340	0.1132	0.1219	0.1070	0.1245
Buston housing	0.1695	0.1708	0.14122	0.1502	0.1403	0.1353	0.1243	0.1379
Auto-MPG	0.1513	0.1584	0.1291	0.1394	0.1321	0.1363	0.1159	0.1279
Stock	0.1380	0.1423	0.1195	0.1245	0.1197	0.1227	0.1084	0.1138
Abalone	0.1327	0.1339	0.1171	0.1179	0.1109	0.1125	0.1077	0.1082
California housing	0.2555	0.2574	0.2265	0.2280	0.1993	0.2035	0.2092	0.2103
Census house (8L)	0.1439	0.1489	0.1254	0.1286	0.1017	0.1023	0.1143	0.1164

Table 4. Comparison of the standard deviation of training and testing (Root Mean Square Error).

	ELM		OELM		I_ELM		NOELM	
	Train	Test	Train	Test	Train	Test	Train	Test
Auto price	0.0033	0.0234	0.0024	0.0215	0.0031	0.0196	0.0018	0.0204
Breast cancer	0.0099	0.0209	0.0088	0.0188	0.0085	0.0167	0.0082	0.0176
Boston housing	0.0130	0.0183	0.0085	0.0148	0.0126	0.0135	0.0059	0.0126
Auto-MPG	0.0142	0.0179	0.0101	0.0141	0.0134	0.0162	0.0077	0.0118
Stock	0.0161	0.0172	0.0125	0.0131	0.0147	0.0158	0.0103	0.0106
Abalone	0.0058	0.0066	0.0053	0.0059	0.0049	0.0056	0.0050	0.0054
California housing	0.0047	0.0061	0.0068	0.0082	0.0027	0.0038	0.0080	0.0094
Census house (8L)	0.0034	0.0038	0.0026	0.0043	0.0006	0.0028	0.0033	0.0046

6. Conclusions

In this paper, referring to the idea of OELM, the orthogonal constraint is introduced into the ELM, then a novel orthogonal ELM is proposed (NOELM), which is a special supervised learning algorithm theoretically. By contrast with the OELM, the main characteristic and contribution is to transform the complex unbalanced orthogonal procrustes problem to a simple least squares problem with orthogonal constraint based on the single vector, and to optimize the single column vector of the output weight matrix so as to obtain the optimal solution of the whole matrix. Compared with ELM and OELM, NOELM can achieve a much better neural network at fast convergence rate and higher training and testing accuracy. Although NOELM is a little weaker than I_ELM in some aspects, the gap is very narrow, and the result is still acceptable.

Author Contributions: L.C. proposed the original idea of the research and wrote some parts of the research. H.Z. carried out the experiments and analyzed the experiments result. H.L. gave related guidance.

Funding: This work was partially support by supported the Fundamental Research Funds for the Central Universities (No.3132019205 and 3132019354), by Liaoning Provincial Natural Science Foundation of China (Grant No.20170520196) and by Scientific Research Funds of Liaoning Provincial educational department (Grant No. JYT2019LQ01 and JYT2019LQ02).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [\[CrossRef\]](#)
- Deo, R.C.; Şahin, M. Application of the Artificial Neural Network model for prediction of monthly Standardized Precipitation and Evapotranspiration Index using hydrometeorological parameters and climate indices in eastern Australia. *Atmos. Res.* **2015**, *161–162*, 65–81. [\[CrossRef\]](#)
- Acharya, N.; Singh, A.; Mohanty, U.C.; Nair, A.; Chattopadhyay, S. Performance of general circulation models and their ensembles for the prediction of drought indices over India during summer monsoon. *Nat. Hazards* **2013**, *66*, 851–871. [\[CrossRef\]](#)
- Deo, R.C.; Tiwari, M.K.; Adamowski, J.F.; Quilty, J.M. Forecasting effective drought index using a wavelet extreme learning machine (W-ELM) model. *Stoch. Environ. Res. Risk Assess.* **2017**, *31*, 1211–1240. [\[CrossRef\]](#)
- Huang, G.-B.; Chen, L. Convex incremental extreme learning machine. *Neurocomputing* **2007**, *70*, 3056–3062. [\[CrossRef\]](#)
- Zhou, Z.; Chen, J.; Zhu, Z. Regularization incremental extreme learning machine with random reduced kernel for regression. *Neurocomputing* **2018**, *321*, 72–81. [\[CrossRef\]](#)
- Wang, D.; Wang, P.; Shi, J. A fast and efficient conformal regressor with regularized extreme learning machine. *Neurocomputing* **2018**, *304*, 1–11. [\[CrossRef\]](#)
- Yin, Y.; Zhao, Y.; Zhang, B.; Li, C.; Guo, S. Enhancing ELM by Markov Boundary based feature selection. *Neurocomputing* **2017**, *261*, 57–69. [\[CrossRef\]](#)
- Ding, X.-J.; Lan, Y.; Zhang, Z.-F.; Xu, X. Optimization extreme learning machine with ν regularization. *Neurocomputing* **2017**, *261*, 11–19.

10. Yildirim, H.; Özkale, M.R. The performance of ELM based ridge regression via the regularization parameters. *Expert Syst. Appl.* **2019**, *134*, 225–233. [[CrossRef](#)]
11. Inaba, F.K.; Salles, E.O.T.; Perron, S.; Caporossi, G. DGR-ELM—Distributed Generalized Regularized ELM for classification. *Neurocomputing* **2018**, *275*, 1522–1530. [[CrossRef](#)]
12. Miche, Y.; Akusok, A.; Veganzones, D.; Björk, K.-M.; Séverin, E.; du Jardin, P.; Termenon, M.; Lendasse, A. SOM-ELM—Self-Organized Clustering using ELM. *Neurocomputing* **2015**, *165*, 238–254. [[CrossRef](#)]
13. Ming, Y.; Zhu, E.; Wang, M.; Ye, Y.; Liu, X.; Yin, J. DMP-ELMs: Data and model parallel extreme learning machines for large-scale learning tasks. *Neurocomputing* **2018**, *320*, 85–97. [[CrossRef](#)]
14. Krishnan, G.S.; S., S.K. A novel GA-ELM model for patient-specific mortality prediction over large-scale lab event data. *Appl. Soft Comput.* **2019**, *80*, 525–533. [[CrossRef](#)]
15. Nayak, D.R.; Zhang, Y.; Das, D.S.; Panda, S. MJaya-ELM: A Jaya algorithm with mutation and extreme learning machine based approach for sensorineural hearing loss detection. *Appl. Soft Comput.* **2019**, *83*, 105626. [[CrossRef](#)]
16. Peng, Y.; Kong, W.; Yang, B. Orthogonal extreme learning machine for image classification. *Neurocomputing* **2017**, *266*, 458–464. [[CrossRef](#)]
17. Peng, Y.; Lu, B.-L. Discriminative manifold extreme learning machine and applications to image and EEG signal classification. *Neurocomputing* **2016**, *174*, 265–277. [[CrossRef](#)]
18. Peng, Y.; Wang, S.; Long, X.; Lu, B.-L. Discriminative graph regularized extreme learning machine and its application to face recognition. *Neurocomputing* **2015**, *149*, 340–353. [[CrossRef](#)]
19. Zhao, H.; Wang, Z.; Nie, F. Orthogonal least squares regression for feature extraction. *Neurocomputing* **2016**, *216*, 200–207. [[CrossRef](#)]
20. Nie, F.; Xiang, S.; Liu, Y.; Hou, C.; Zhang, C. Orthogonal vs. uncorrelated least squares discriminant analysis for feature extraction. *Pattern Recognit. Lett.* **2012**, *33*, 485–491. [[CrossRef](#)]
21. Zhang, Z.; Du, K. Successive projection method for solving the unbalanced Procrustes problem. *Sci. China Ser. A* **2006**, *49*, 971–986. [[CrossRef](#)]
22. Bache, K.; Lichman, M. UCI Machine Learning Repository. University of California, School of Information and Computer Sciences: Irvine, CA, USA, 2013. Available online: <http://archive.ics.uci.edu/ml> (accessed on 11 October 2019).
23. Xu, Z.; Yao, M.; Wu, Z.; Dai, W. Incremental Regularized Extreme Learning Machine and It's Enhancement. *Neurocomputing* **2015**, *174*, 134–142. [[CrossRef](#)]
24. Huang, G.-B.; Chen, L.; Siew, C.-K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* **2006**, *17*, 879–892. [[CrossRef](#)] [[PubMed](#)]
25. Ying, L. Orthogonal incremental extreme learning machine for regression and multiclass classification. *Neural Comput. Appl.* **2016**, *27*, 111–120. [[CrossRef](#)]

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).