A New Learning Algorithm for Function Approximation Incorporating *A Priori* Information into Extreme Learning Machine

Fei Han[1,2], Tat-Ming Lok[3], and Michael R. Lyu[4]

[1] Intelligent Computing Lab, Hefei Institute of Intelligent Machines,
Chinese Academy of Sciences, P.O. Box 1130, Hefei, Anhui 230031, China
[2] Department of Automation,
University of Science and Technology of China, Hefei 230027, China
[3] Information Engineering Dept., The Chinese University of Hong Kong, Shatin, Hong Kong
[4] Computer Science & Engineering Dept.,
The Chinese University of Hong Kong, Shatin, Hong Kong
hanfei1976@iim.ac.cn, tmlok@ie.cuhk.edu.hk,
lyu@cse.cuhk.edu.hk

Abstract. In this paper, a new algorithm for function approximation is proposed to obtain better generalization performance and faster convergent rate. The new algorithm incorporates the architectural constraints from *a priori* information of the function approximation problem into Extreme Learning Machine. On one hand, according to Taylor theorem, the activation functions of the hidden neurons in this algorithm are polynomial functions. On the other hand, Extreme Learning Machine is adopted which analytically determines the output weights of single-hidden layer FNN. In theory, the new algorithm tends to provide the best generalization at extremely fast learning speed. Finally, several experimental results are given to verify the efficiency and effectiveness of our proposed learning algorithm.

1 Introduction

Most traditional learning algorithms with feedforward neural networks (FNN) are to use backpropagation (BP) algorithm to derive the updated formulae of the weights [1]. However, these learning algorithms have the following major drawbacks that need to be improved. First, they are apt to be trapped in local minima. Second, they have not considered the network structure features as well as the involved problem properties, thus their generalization capabilities are limited [2-7]. Finally, since gradient-based learning is time-consuming, they converge very slowly [8-9].

In literatures [10-11], a learning algorithm was proposed that is referred to as Hybrid-I method. In this algorithm, the cost terms for the additional functionality based on the first-order derivatives of neural activation at hidden layers were designed to penalize the input-to-output mapping sensitivity. In literature [12], a modified hybrid learning algorithm (MHLA) was proposed according to Hybrid-I algorithm to improve the generalization performance. Nevertheless, it was found from the experimental results that the computational requirements for the above two algorithms

J. Wang et al. (Eds.): ISNN 2006, LNCS 3971, pp. 631 – 636, 2006.

are actually relatively large. These learning algorithms can almost improve the generalization performance to some degree, but there is not the best one resulted.

In literature [13], the relations between the single-hidden layer FNN (SLFN) and the corresponding hidden layer to output layer network were lucubrated. In literatures [8-9], a learning algorithm for SLFN which was called as Extreme Learning Machine (ELM) was proposed. ELM randomly chooses the input weight and analytically determines the output weights of SLFN through simple generalized inverse operation of the hidden layer output matrices. Therefore, ELM has better generalization performance with much faster learning speed. However, ELM also had not considered the network structure features as well as the involved problem properties and its generalization performance is also limited to some extent.

In this paper, a new learning algorithm for function approximation problem incorporating *a priori* information into ELM is proposed. The new learning algorithm selected the hidden neurons activation functions as polynomial functions on the basis of Taylor series expansion. Moreover, the new algorithm analytically determines the output weights of SLFN through simple generalized inverse operation of the hidden layer output matrices according to ELM. Finally, theoretical justification and simulated results are given to verify the better generalization performance and faster convergent rate of the proposed constrained learning algorithm.

2 Extreme Learning Machine

In order to find an effective solution to the problem caused by BP learning algorithm, Huang [8-9] proposed ELM. Since a feedforward neural network with single nonlinear hidden layer is capable of forming an arbitrarily close approximation of any continuous nonlinear mapping, the ELM is limited to such networks.

For N arbitrary distinct samples (x_i, t_i), where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in R^n$, $t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in R^m$. The SLFN with H hidden neurons and activation function $g(x)$ can approximate these N samples with zero error means that

$$\mathbf{H}\, wo = \mathbf{T} \tag{1}$$

where

$$\mathbf{H}(wh_1, wh_2, \ldots, wh_H, b_1, b_2, \ldots, b_H, x_1, x_2, \ldots, x_N)$$

$$= \begin{bmatrix} g(wh_1 x_1 + b_1) & \cdots & g(wh_H x_1 + b_H) \\ \vdots & \cdots & \vdots \\ g(wh_1 x_N + b_1) & \cdots & g(wh_H x_N + b_H) \end{bmatrix}_{N \times H}, wo = \begin{bmatrix} wo_1^T \\ \vdots \\ wo_H^T \end{bmatrix}_{H \times m}, \mathbf{T} = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \tag{2}$$

where $wh_i = [wh_{i1}, wh_{i2}, \ldots, wh_{in}]^T$ is the weight vector connecting the ith hidden neuron and the input neurons, $wo_i = [wo_{i1}, wo_{i2}, \ldots, wo_{im}]^T$ is the weight vector connecting the ith hidden neuron and the output neurons, and b_i is the threshold of the ith hidden neuron. In order to make it easier to understand ELM, a theorem is introduced in the following:

Theorem 2.1 [14]. Let there exist a matrix G such that Gy is a minimum norm least-squares solution of a linear system $Ax = y$. Then it is necessary and sufficient that $G = A^+$, the Moore-Penrose generalized inverse of matrix A.

In the course of learning, first, the input weights wh_i and the hidden layer biases b_i are arbitrarily given and need not be adjusted at all. Second, according to Theorem 2.1, the smallest norm least-squares solution of the above linear Eqn. (1) is as follow:

$$wo = \mathbf{H}^{+}\mathbf{T} \tag{3}$$

From the above discussion, it can be found that the ELM has the minimum training error and smallest norm of weights. The smallest norm of weights tends to have the best generalization performance. Since the smallest norm least-squares solution of the above linear Eqn. (1) is obtained by analytical method and all the parameters of SLFN need not to be adjusted, ELM converge much faster than BP algorithm.

3 New Learning Algorithm Incorporating a Priori Information into ELM

3.1 Architectural Constraints from a Priori Information

According to the Taylor theorem, if the function meets the conditions that the Taylor theorem requires, the function has the corresponding Taylor expansion as follows:

$$f(x) = f(x_0) + \sum_{k=1}^{n} \frac{x^k}{k!} f^{(k)}(x_0) + \frac{x^{(n+1)}}{(n+1)!} f^{(n+1)}(\xi),\ x, x_0, \xi \in D(f(x)),\ \xi \in (x, x_0) \text{ or } \xi \in (x_0, x). \tag{4}$$

where $D(f(x))$ denotes the definitional domain of the function $f(x)$.

From Eqn.(4), it can be found that the function which meets the conditions of the Taylor theorem can be expressed as the weighted sum of the polynomial functions. In order to approximate the function $f(x)$ more accurately by the FNN $\phi(x)$, we make the FNN $\phi(x)$ be expressed as the weighted sum of the polynomial functions according to the above *a priori* information. So a SLFN is adopted for approximating the function and the transfer function of the k th hidden neuron is selected as the function $x^k / k!$, $(k = 1, 2, \ldots, n.)$. Then the FNN $\phi(x)$ can be expressed as follows:

$$\phi(x) = \sum_{k=1}^{n} wo_k \frac{(wh_k x)^k}{k!} - wo_{n+1}, \tag{5}$$

where wo_k denotes the the synaptic weight from the output neuron to the k th neuron at the hidden layer, and wh_k denotes the the synaptic weight from the k th neuron at the hidden layer to the input neuron. The output layer is a linear neuron.

3.2 New Learning Algorithm

In order to improve the generalization performance and obtain faster convergent rate, a new algorithm incorporating *a priori* information into ELM is proposed as follows:

First, according to Subsection 3.1, a SLFN as shown in Eqn. (5) is adopted for approximating the function. The weights from the input layer to the hidden layer are all fixed to one, i.e., $wh_k = 1, k = 1,2,\cdots,n.$ According to Section 2, the weights from the output neuron to the hidden neurons are analytically determined by Eqn. (3).

In the new algorithm, the weights from the output neuron to the hidden neurons are analytically determined, so the learning speed of the new algorithm can be thousands of times faster than that of BP algorithm. Moreover, according to Eqn. (3), since the smallest norm least-squares solution is obtained, the new algorithm tends to have the better generalization performance. Finally, compared with ELM, in that the new learning algorithm incorporates architectural constraints from *a priori* information into SLFN, the new learning one has better generalization performance than ELM. From this new algorithm, the following conclusion can be easily deduced:

Conclusion 1. Assume that the FNN, $\phi(x)$, which is expressed as Eqn. (5), is used to approximate the function $f(x)$ by the above new learning algorithm. The function $f(x)$ meets the conditions that the Taylor theorem requires and $0 \in D(f(x))$. The following equation can be obtained:

$$wo_k \approx f^{(k)}(0),\ k = 1,2,\cdots,n. \qquad wo_{n+1} \approx -f(0) \tag{6}$$

Proof. Comparing Eqn.(6) and Eqn.(7), we notice that $f(x) \approx \phi(x)$ and $wh_k = 1, (k = 1,2,\cdots,n)$ from the new learning algorithm. Therefore, Eqn. (8) can be easily deduced. Q.E.D.

4 Experimental Results

To demonstrate the improved generalization performance and fast convergent rate of the new learning algorithm, in the following we shall conduct the experiments with two functions. They are a bimodal function $y = \sin(2x)/2$ and a multimodal function $y = (1 - (40x/\pi) + 2(40x/\pi)^2 - 0.4(40x/\pi)^3)e^{-x/2}$. In this section, this new algorithm is compared with traditional BP algorithm, Hybrid-I algorithm, MHLA and ELM. The activation function of the neurons in all layers for BP algorithm, Hybrid-I algorithm and MHLA all are tangent sigmoid function. The activation functions of the hidden neurons for ELM are sigmoid function. In all five learning algorithms, the number of the hidden neurons is 10. As for each function, assume that 126 training samples are selected from [0,π] at identical spaced interval. Likely, 125 testing samples are also selected from [0.0125, π-0.0125] at identical spaced interval.

In order to statistically compare the approximation accuracies and CPU time for the two functions with the above five algorithms, we conducted the experiments fifty times for each algorithm, and the corresponding results are summarized in Table 1-2.

Table 1. The approximation accuracies and CPU time for $y = \sin(2x)/2$ with the five algorithms

LA	Training error	Testing error	CPU time
BP	1.2956e-5	1.1925e-5	53.5160s
Hybrid-I	5.0663e-6	5.0400e-6	65.7138s
MHLA	2.6359e-6	2.5472e-6	75.5460s
ELM	5.3231e-11	4.8595e-11	0.0631s
New LA	2.3636e-12	2.1346e-12	0.2020s

Table 2. The approximation accuracies and CPU time for $y = (1-(40x/\pi)+2(40x/\pi)^2-0.4(40x/\pi)^3)e^{-x/2}$ with the five algorithms

LA	Training error	Testing error	CPU time
BP	5.2511e-4	4.5036e-4	54.0630s
Hybrid-I	2.6711e-4	2.1255e-4	75.5628s
MHLA	1.5123e-4	1.1102e-4	85.8660s
ELM	5.3450e-6	5.1332e-6	0.0825s
New LA	1.4665e-6	1.0535e-6	0.3523s

From the above results, it can be drawn the conclusions as follows:

First, the generalization performance of the new algorithm and ELM is much better than that of the BP algorithm, Hybrid-I algorithm and MHLA, because the testing error of the new algorithm and ELM is much less than that of other three algorithms. This result rests in the fact that the new algorithm and ELM obtain the smallest norm least-squares solution through Eqn. (3), whereas other three algorithms do not.

Second, the new algorithm and ELM converge much faster that the BP algorithm, Hybrid-I algorithm and MHLA. This is because the new learning algorithm and ELM obtain the solution by analytical method, whereas other three algorithms obtain the solution through thousands of iterative calculation.

Third, compared with ELM, the new algorithm has better generalization. This is chiefly because the new learning one considers *a priori* information from the function approximation problem.

Finally, compared with ELM, the new learning algorithm converges slightly slower than ELM. This rests in the fact that the new learning one requires much more time to calculate the hidden neurons outputs than ELM.

5 Conclusions

In this paper, a new learning algorithm which incorporates the architectural constraints into ELM was proposed for function approximation problem. The architectural constraints are extracted from *a priori* information of the approximated function based on Taylor series expansion. The architectural constraints are realized by selecting the activation functions of the hidden neurons as polynomial functions. Furthermore, the new algorithm analytically determines the output weights of SLFN through simple generalized inverse operation of the hidden layer output matrices

according to ELM. Therefore, the new learning one has much better generalization performance and faster convergent rate than the traditional gradient-based learning algorithms. Finally, theoretical justification and simulated results were given to verify the efficiency and effectiveness of the proposed new learning algorithm. Future research works will include how to apply this new learning algorithm to resolve more numerical computation problems.

Acknowledgement

This work was supported by the National Science Foundation of China (Nos.60472111, 30570368 and 60405002).

References

1. Ng, S.C., Cheung, C.C., Leung, S.H.: Magnified Gradient Function with Deterministic Weight Modification in Adaptive Learning, IEEE Transactions on Neural Networks 15(6) (2004) 1411-1423
2. Baum, E., Haussler, D.: What Size Net Gives Valid Generalization? Neural Comput. 1(1) (1989) 151-160
3. Huang, D.S.: A Constructive Approach for Finding Arbitrary Roots of Polynomials by Neural Networks, IEEE Transactions on Neural Networks 15(2) (2004) 477-491
4. Huang, D.S., Chi, Z.: Finding Roots of Arbitrary High Order Polynomials Based on Neural Network Recursive Partitioning Method, Science in China Ser. F Information Sciences 47(2) (2004) 232-245
5. Huang, D.S., Ip, Horace H.S., Chi, Z.: A Neural Root Finder of Polynomials Based on Root Moments, Neural Computation 16(8) (2004) 1721-1762
6. Huang, D.S., Ip, Horace H.S., Chi, Z., Wong, H.S.: Dilation Method for Finding Close Roots of Polynomials Based on Constrained Learning Neural Networks, Physics Letters A 309 (5-6) (2003) 443-451
7. Karras, D.A.: An Efficient Constrained Training Algorithm for Feedforward Networks, IEEE Trans. Neural Networks 6(6) (1995) 1420-1434
8. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme Learning Machine: A New Learning Scheme of FNN, 2004 International Joint Conference on Neural Networks (IJCNN'2004), July 25-29, Budapest, Hungary, 985-990
9. Huang, G.B., Siew, C.K.: Extreme Learning Machine with Randomly Assigned RBF Kernels, International Journal of Information Technology 11(1) (2005), 16-24
10. Jeong, S.Y., Lee, S.Y.: Adaptive Learning Algorithms to Incorporate Additional Functional Constraints into Neural Networks, Neurocomputing 35 (1-4) (2000), 73-90
11. Jeong, D.G., Lee, S.Y.: Merging Back-propagation and Hebbian Learning Rules for Robust Classifications, Neural Networks 9(7) (1996) 1213-1222
12. Han, F., Huang, D.S., Cheung, Y.-M., Huang, G.B.: A New Modified Hybrid Learning Algorithm for FNN, Lecture Notes in Computer Science, Vol. 3496, Springer-Verlag (2005) 572-577
13. Huang, D.S.: Systematic Theory of Neural Networks for Pattern Recognition, Publishing House of Electronic Industry of China, Beijing, 1996, 109-110
14. Serre, D.: Matrices: Theory and Application, Springer-Verlag (2002) 147-147