Available online at www.sciencedirect.com

SciVerse ScienceDirect

AASRI Procedia 3 (2012) 375 – 380

AASRI Procedia

www.elsevier.com/locate/procedia

2012 AASRI Conference on Modelling, Identification and Control

Hidden Node Optimization for Extreme Learning Machine

Yan-wei Huang*, Da-hu Lai

College of Electrical Engineering & Automation, Fuzhou University, Fuzhou, Fujian, China

Abstract

The number of hidden nodes is a critical factor for the generalization of ELM. Generally, it is heavy for time consumption to obtain the optimal number of hidden nodes with trial-and-error. A novel algorithm is proposed to optimize the hidden node number to guarantee good generalization, which employs the PSO in the optimization process with structural risk minimization principle. The simulation results indicate our algorithm for the optimal number of hidden nodes is reasonable and feasible with 6 datasets on benchmark problems by the accuracy comparisons.

Selection and/or peer review under responsibility of American Applied Science Research Institute

Keywords: ELM, VC confidence, structural risk, hidden nodes, PSO;

1. Introduction

Extreme Learning Machine(ELM)[1], a novel learning algorithm with fast learning speed and good generalization, is a single-hidden-layer feed forward neural networks(SLFNs). As traditional SLFNs, ELM tends to achieve a good generalization with the suitable hidden nodes. EM-ELM[2] pointed that how to choose the optimal number of hidden nodes is still unknown and important. RCGA-ELM[3] employs genetic algorithm (GA) to optimize the number of hidden nodes, input weights(w) and bias(b) in five-fold cross-validation procedures. It is so complex and time-consuming by involving multiple operators. PSO-ELM[4] optimizes only the weight w and bias b, illustrating their value ranges, Root mean square error(RMSE) as the

* *HUANG Yan-Wei.Tel: 059122866596; fax: ;
E-mail Address: sjtu_huanghao@sina.com

Selection and/or peer review under responsibility of American Applied Science Research Institute
doi:10.1016/j.aasri.2012.11.059

optimization goals and variance(δ_{RMSE}) as the iterative stop criteria. ICGA-SRM-ELM[5] only considers RMSE and hidden nodes, doesn't give a specific form of the solution. Miche[6] presented OP-ELM to choose hidden nodes automatically. However, it needs Multi-response Sparse Regression and Leave-One-Out algorithms to get rid of the useless neurons of the hidden layer, which has to employ multi-certeria mechanism to increase or decrease the hidden node, tending to be more complex. μG-ELM[7] as another solution to optimize the hidden nodes by GA is easy to overfit with RMSE as the only training goal. Those previous works have made some attempts to improve generalization of ELM, but existed some shortcomings like overfitting, heavy time consumption or without specific form of the objective function.

SRM-ELM (Structural Risk Minimization ELM, SRM-ELM) algorithm is proposed in this work to obtain an optimal number of hidden nodes for ELM by PSO, with Structural Risk Minimization (SRM) principle that consist of empirical risk and VC confidence. The most superiority of SRM-ELM is to avoid the overfitting by introduced the VC theory. Moreover, PSO chosen as the optimal tool will reduce the operation time compared with GA or DE (Differential Evolution) algorithm.

2. Basic of ELM

Given N learning samples $[x_i,y_i]$, where $x_i=[x_{i1},x_{i2},\ldots,x_{in}]^T\in R^n$, $y_i=[t_{i1},t_{i2},\ldots,t_{im}]^T\in R^m$ and $i=1,2,\ldots,N$, the SLFN is constructed with L hidden notes and the activation function $g(w,x,b)$. ELM modeling that is looking for a function $f(x)$ to obtain the right output y with respect to x (outside of the samples) after training network with $[x_i,y_i]$. $f(x)$ is defined as[8] ,

$$y=f(x)=\sum_{i=1}^{L}\beta_i g(w_i,x,b_i) \tag{1}$$

where $\beta_i=[\beta_{i1},\beta_{i2},\ldots,\beta_{im}]^T$ is the output weights vector connecting the ith hidden node and the output nodes, $w_i=[w_{i1},w_{i2},\ldots,w_{in}]^T$ is the input weights vector connecting the ith hidden node and the input nodes, b_i is the threshold of the ith hidden node, $i=1,2,\ldots,L$.

According to N samples $[x_i,y_i]$, Eq.(1) can be further written as

$$H\beta=Y \tag{2}$$

where

$$H(w_1,\ldots,w_L,b_1,\ldots,b_L,x_1,\ldots,x_N)=\begin{bmatrix} g(w_1,x_1,b_1) & \cdots & g(w_L,x_1,b_L) \\ \vdots & \cdots & \vdots \\ g(w_1,x_N,b_1) & \cdots & g(w_L,x_N,b_L) \end{bmatrix}_{N\times L} \quad \beta=\begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L\times m} \quad Y=\begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N\times m} \tag{3}$$

As in [8], H is the hidden layer output matrix of the neural network. For $L\ll N$, H is a non-square matrix, given any w and b, H^- can be obtained according to Moore-Penrose generalized inverse theorem, β is

$$\beta=H^-Y \tag{4}$$

With Eq.(4), β can be gained after setting parameters of ELM (hidden nodes L, activation function $g(x)$, any data w and b) to accomplish the training of ELM.

3. Optimization of hidden nodes for ELM

Given the w, b and activation function, the generalization performance mainly relies on hidden nodes. Here, we present SRM-ELM to achieve the optimal number of hidden nodes by PSO based on SRM.

3.1. Basic of SRM

Some defects happen to Empirical Risk Minimization (ERM) principle has some disadvantages in machine

learning, so Vapnik presented SRM to improve the generalization performance. Consider a set of finite function $0 \le Q(z,\alpha) \le B$, $\alpha \in \Lambda$, while the VC-dimension h is limited[9], set Eq.(5) with probability at least 1-η,

$$R(\alpha) \le R_{emp}(\alpha) + \frac{B\varepsilon}{2}\left(1+\sqrt{1+\frac{4R_{emp}(\alpha)}{B\varepsilon}}\right) \tag{5}$$

where $R_{emp}(\alpha)$ is an empirical risk, and

$$\varepsilon = a\frac{h(\ln(bn/h)+1)-\ln(\eta/4)}{n} \tag{6}$$

where n is the sample number, $0 < a \le 4$, $0 < b \le 2$, $\eta \in (0,1]$,and B=1 in binary-class issues. The second summand on the right hand side of Eq.(5) is called VC confidence, whose value depends on VC-dimension h in the case of given samples. According to SRM, minimal $R(\alpha)$ will be obtained by minimizing right hand side of Eq.(5).

3.2. Modification for SRM

VC-dimension h can be used as a measure of the computational complexity for machine learning. A good VC-dimension can improve the generalization of neural network. But, there is not a universal formula to calculate h for neural network. VC-dimension h was deduced with some formulae for the feedforward network with sigmoid activation function[10-12],

$$h_{\text{lower bound}} = \lambda^2, \quad h_{\text{lower bound}} = \lambda l, \quad h_{\text{upper bound}} = \lambda^2 n^2 \tag{7}$$

where λ is the number of weights in network, l is the number of layers and n is the sum of hidden nodes and output nodes. Some attention could be took on Eq.(7) that h has some relation to λ, and λ also has some relation to the total number of nodes. So, Eq.(7) suppose h can be obtained from the total number of nodes in network. Here, we consider the total number of nodes as the VC-dimension h for ELM. h can be written as,

$$h = input\_nodes + hidden\_nodes + output\_nodes \tag{8}$$

According to statistic learning theory, a=4, b=1 in Eq.(6), then Eq.(6) can be rewritten as,

$$\varepsilon_1 = 4\frac{h(\ln(n/h)+1)-\ln(\eta/4)}{n} \tag{9}$$

B=1 in Eq.(5), VC confidence in Eq.(5) can be revised as,

$$f(h) = \frac{\varepsilon_1}{2}\left(1+\sqrt{1+\frac{4R_{emp}(\alpha)}{\varepsilon_1}}\right) \tag{10}$$

Then differentiation to h of Eq.(10) is ,

$$f'(h) = \frac{2\ln(n/h)}{n}\left(1+\sqrt{1+\frac{4R_{emp}(\alpha)}{\varepsilon_1}} - \frac{2R_{emp}(\alpha)}{\varepsilon_1\sqrt{1+\frac{4R_{emp}(\alpha)}{\varepsilon_1}}}\right) \tag{11}$$

With Eq.(11), $f(h)$ tends to be maximum value when h=n. Generally, VC confidence should be concave function. However, we find that Eq.(10) is a convex function with respect to $h \in [0, n]$ in Fig.1. So VC confidence should be transformed into another form.

It is obvious that Eq.(12) is true,

$$\phi(h) = e^{f(h)} > f(h) \tag{12}$$

then Eq.(5) is rewritten as,

$$R(\alpha) \le R_{emp}(\alpha) + f(h) < R_{emp}(\alpha) + \phi(h) \tag{13}$$

Eq.(5) can be transformed,

$$P\{R(\alpha) \le R_{emp}(\alpha) + f(h)\} \ge 1\text{-}\eta \tag{14}$$

With Eq.(13) and Eq.(14),

$$P\{R(\alpha) \le R_{emp}(\alpha) + \phi(h)\} \ge \eta_0 > 1\text{-}\eta \tag{15}$$

so set Eq.(16) with probability at least 1-η,

$$R(\alpha) < R_{emp}(\alpha) + \phi(h) \tag{16}$$

where $\eta_0 \in (0,1]$. We do simulations on binary-class issue Haberman from UCI database to evaluate $\phi(h)$ in Fig.2. Obviously, $\phi(h)$ is a concave function, but $f(h)$ is a convex function. So Eq.(17) can be used as the objective function, which consist of R_{emp} and $\phi(h)$.

$$Objectfun: F(\alpha, h) = R_{emp}(\alpha) + \phi(h) \tag{17}$$

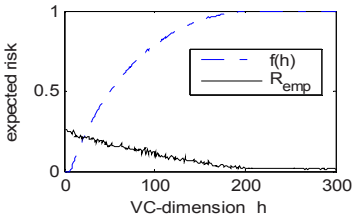

Fig.1 Relationship of $f(h)$ and R_{emp}

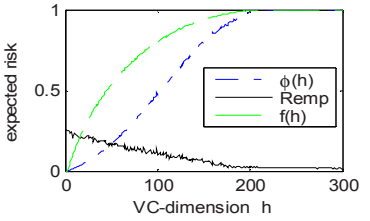

Fig.2 Relationship of $f(h)$, $\phi(h)$ and R_{emp}

3.3. PSO algorithm

PSO is used to optimize the objective function Eq.(17). We define the particles position p as the hidden node number L. The update rules for the position p and velocity v are,

$$v_{id}^{k+1} = \tau * v_{id}^{k} + c_1 * rand_1 * (pbest_{id}^{k} - p_{id}^{k}) + c_2 * rand_2 * (gbest_{d}^{k} - p_{id}^{k}) \tag{18}$$

$$p_{id}^{k+1} = p_{id}^{k} + v_{id}^{k+1} \tag{19}$$

where p_{id}^{k} is the dth position of the ith particle in the kth iteration, and v_{id}^{k} is the dth velocity of the ith particle in the kth iteration. $pbest_{id}^{k}$ is the dth best position of the ith particle in the kth iteration, and $gbest_{d}^{k}$ is the dth best position of the population in the kth iteration. d=1,2,…N. c_1 and c_2 are the learning factors, usually equal to constant 2. $rand_1$ and $rand_2$ are random number in the range of (0,1). Formula of Inertia weight τ is, $\tau = \tau_{\max} - \frac{\tau_{\max} - \tau_{\min}}{it_{\max}} * N_C$ where τ_{max} and τ_{min} represent the maximum and minimum value of inertia weight respectively. it_{max} represents the maximum value of iteration, and N_C is the current iteration. Particles have a minimum and a maximum velocity, also a minimum and a maximum position, the rules are in PSO-ELM[4].

3.4. SRM-ELM Algorithm flow

Fig.3 is the flow of SRM-ELM algorithm. After initialize swarm population, the position value p is the hidden node number L, which is the key link between PSO and ELM. p and v will update in every iteration. If $N_C > it_{\max}$, output the optimal number of hidden nodes, else repeat the flow after initialization.

4. Experimental Results

4.1. Parameters

Parameters setting for PSO: population size: n=50, maximum iteration: it_{max}=30, learning factors: c_1= c_2=2, dimension: d=1, the maximum and minimum value of inertia weight: τ_{max}=0.9, τ_{min}=0.4, range of the position(number of hidden nodes): [X_{min},X_{max}]=[1,300], range of the velocity: [V_{min},V_{max}]= [-(X_{max}-X_{min}),(X_{max}-X_{min})]=[-299,299]. Parameters for ELM: choose the sigmoid for the activation function. Parameters for objective function $F(\alpha,h)$:a=4,b=1 for Eq.(6), $\eta=\sqrt{1/n}$, where n is the training sample number.

4.2. Result comparisons

All the simulations for SRM-ELM are carried out in Matlab7.0 in environment running in a Pentium(R) Dual-core 3.19GHZ CPU. We apply 6 datasets from UCI database to test the performance of SRM-ELM. Table.1 is the description for 6 datasets.

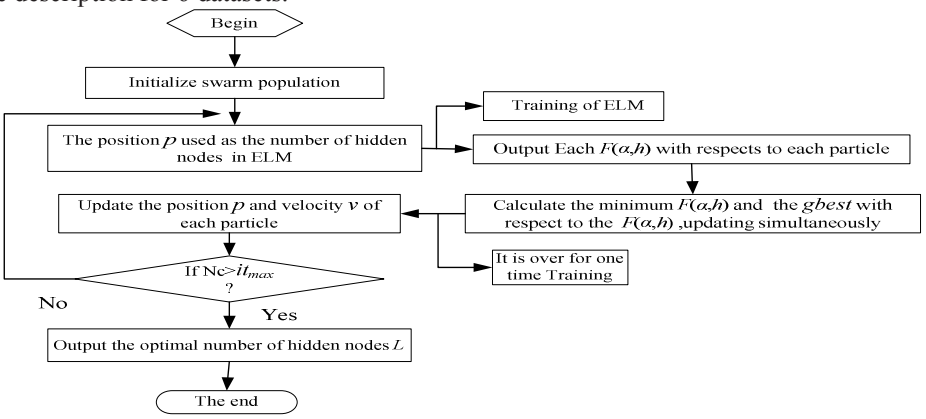

Fig.3 Flow for SRM-ELM algorithm

Table.1 the description for 6 datasets

Datasets	Classes		Attributes	Training data	Testing data
Haberman	2	3	230	79	
Blood	2	4	548	200	
Pima	2	8	400	368	
Ionosphere	2	34	250	101	
Breast Cancer	2	9	500	199	
Australian Credit	2	14	500	190	

The classification accuracy and the number of hidden nodes are used to evaluate the performance for SRM-ELM. We also compare SRM-ELM with the other two algorithms, one is the original ELM with 10-fold cross validation, the other one is using the cut-and-try work $N=\sqrt{a+b}+c$ to select the node number for ELM, where N is the number of hidden nodes, a is the input nodes, b is the output nodes, and c is the random number in 1~10.

Table.2 is the result comparisons for accuracy and the node number. The optimal number by SRM-ELM drops in the range by ELM, and the accuracy of SRM-ELM are close to ELM's, which indicates that SRM-ELM is feasible and effective. Moreover, the performance of SRM-ELM is better than the $N=\sqrt{a+b}+c$ in most of the datasets.

5. Conclusions

This work proposed a novel algorithm to optimize the number of hidden nodes for ELM by SRM and PSO. We modified the formula for the VC confidence to reconstruct a concave function for SRM as the objective function. Then we employed PSO to optimize the SRM function for the optimal number for hidden nodes for ELM. The experiment results demonstrate that our algorithm can be used to obtain the effective number of hidden nodes and an excellent generalization.

Table.2 Performance Comparison in SRM-ELM, ELM and $N = \sqrt{a+b} + c$

Datasets	SRM-ELM		ELM		$N = \sqrt{a+b} + c$	
	L	accuracy(%)	Range of L	accuracy(%)	Range of L	accuracy(%)
Haberman	7±1	0.7386	4~9	0.7395	3~12	0.7381
Blood	17±3	0.8673	15~45	0.8692	3~12	0.8613
Pima	17±2	0.7909	10~17	0.7936	4~13	0.7918
Ionosphere	25±5	0.9040	25~55	0.9109	7~16	0.8564
Breast Cancer	15±6	0.9945	10~50	0.9960	4~13	0.9960
Australian Credit	17±3	0.8653	13~23	0.8653	5~14	0.8518

Note: L is the number of hidden nodes

Acknowledgements

This work is sponsored by Doctoral Fund of Chinese Ministry of Education(20113514120007) and Nature Science Fund of Fujian Province in China(2010J05132).

References

[1] G.-B.Huang, Q.-Y.Zhu, C.-K.Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. Proc of the IEEE International Joint Conference on Neural Networks. 2004; 2: 985–990.
[2] Guorui Feng, GuangBin Huang, Qingping Lin. Error Minimized Extreme Learning Machine With Growth of Hidden Nodes and Incremental Learning. IEEE Trans on Neural Networks, 2009; 20(8):1352-1357.
[3] S.Suresh, S.Saraswathi, N.Sundararajan. Performance Enhancement of Extreme Learning Machine for Multi-Category Sparse Cancer Classification. Engineering Applications of Artificial Intelligence, 2010; 23(7): 1149-1157.
[4] You Xu, Yang Shu. Evolutionary Extreme Learning Machine Based on Particle Swarm Optimization. Lecture Notes in Computer Science Advances in Neural Networks, 2006; 3971:64-652.
[5] Saraswathi, S.Sundaram, S.Sundararajan.N. ICGA-PSO-ELM Approach for Multiclass Cancer Classification Resulting Reduced Gene Sets in Which Genes Encoding Secreted Proteins Are Highly Represented inaccurate. IEEE/ACM Trans on Computational Biology and Bioinformatics, 2010; 8(2):452-463.
[6] Yoan Miche, Sorjamaa.A, Bas.P. OP-ELM:Optimally Pruned Extreme Learning Machine. IEEE Trans on Neural Networks, 2010; 21(1):158-162.
[7] Lahoz.D, Lacruz.B, Mateo.P.M. A Bi-objective Micro Genetic Extreme Learning machine. IEEE workshop on Hybrid Intelligent Models And Applications; 2011, 68-75.
[8] GuangBin Huang, QinYu Zhu, CheeKheong Siew. Extreme learning machine: Theory and applications. Neurocomputing, 2006; 70:489-501
[9] Vladimir N. Vapnik. Statistical Learning Theory. 2nd ed. New York: Wiley; 1998.
[10] Koiran P, Sontag E D. Neural Networks with Quadratic VC Dimension. Journal of Computer and System Sciences. 1997; 54:190-198.
[11] Bartlett P L, Maiorov V, Meir R. Almost Linear VC-Dimension Bounds for Piecewise Polynomial Networks. Neural Computation.1998; 10(8):2159-2173.
[12] Karpinski M, Macintyre A. Polynomial Bounds for VC Dimension of Sigmoidal and General Pfaffian Neural Networks. Journal of Computer and System Sciences.1997; 54:169-176.