Journal of Computer and Communications, 2015, 3, 191-197
Published Online November 2015 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2015.311030

Mutual Information-Based Modified Randomized Weights Neural Networks

Jian Tang[1,2], Zhiwei Wu[2], Meiying Jia[1], Zhuo Liu[2]

[1]Research Institute of Computing Technology, Beifang Jiaotong University, Beijing, China
[2]State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China
Email: powernature@126.com

Received October 2015

Abstract

Randomized weights neural networks have fast learning speed and good generalization performance with one single hidden layer structure. Input weighs of the hidden layer are produced randomly. By employing certain activation function, outputs of the hidden layer are calculated with some randomization. Output weights are computed using pseudo inverse. Mutual information can be used to measure mutual dependence of two variables quantitatively based on the probability theory. In this paper, these hidden layer's outputs that relate to prediction variable closely are selected with the simple mutual information based feature selection method. These hidden nodes with high mutual information values are maintained as a new hidden layer. Thus, the size of the hidden layer is reduced. The new hidden layer's output weights are learned with the pseudo inverse method. The proposed method is compared with the original randomized algorithms using concrete compressive strength benchmark dataset.

Keywords

Randomized Weights Neural Networks, Mutual Information, Feature Selection

1. Introduction

Machine learning (ML)-based data analysis has been a hot focuses in different disciplines. The most used learning prediction model construction methods are backup propagation neural networks (BPNN) and support vector machines (SVM) [1]. However, BPNN suffers from local optima, uncontrolled convergence speed and over-fitting problems. Although SVM can address small samples modeling problem with good generalization, quadratic program (QP) and large kernel matrix problems are difficult to overcome for big sample learning datasets. A special single-layer feed-forward (SLFN) networks-based neural networks learning algorithm, *i.e.*, randomized weights neural networks, was proposed to overcome shortcomings that caused by the gradient-based learning algorithms [2] [3]. Its characteristics include: 1) input weights of the hidden layer are chosen randomly; 2) the hidden layer neurons need not be adjusted; and 3) output weights are analytically computed using pseudo inverse or least square method. The normally used pseudo inverse-based output weights calculation method has

How to cite this paper: Tang, J., Wu, Z.W., Jia, M.Y. and Liu, Z. (2015) Mutual Information-Based Modified Randomized Weights Neural Networks. *Journal of Computer and Communications*, **3**, 191-197.
http://dx.doi.org/10.4236/jcc.2015.311030

two advantages: a) optimal solution to the least square problem can be obtained; and b) the optimal output weight matrix is with minimal norm. There, this randomized weights neural networks algorithm has faster learning speed, which has been successfully applied [4] [5]. Thus, pseudo inverse-based randomized algorithm solves the local minima problem with good testing performance and fast training time [6]. However, how to control and estimate randomization of the input weights is an open issue. Study shows that small norm of the weights is more important than the node number to obtain good generalization performance for feed forward networks [7]. The norms of the hidden weights generated by deep learning are small [8]. Therefore, a randomized algorithms for nonlinear system identification with deep learning modification is proposed, which regards deep learning as pre-training technique to obtain the hidden layers' input weights [9]. Thus, the small norm of the input weights and output weights are obtained by combination of the deep learning and the least-square approaches. However, long training time is needed. An effective and simple randomization control and estimation method needs to be addressed further.

Mutual information (MI) can be used to measure the mutual dependence of the two variables quantitatively based on the probability theory and information theory. Thus, it has been used widely in feature selection. The MI is more comprehensive than the other normal feature selection methods for select optimal input variables [10]. However, the popular used MI based feature selection method needs lots of computational consume [11]. A simple MI based feature selection method is used in [12] [13]. For randomized weights neural networks, if we cannot control the randomization of the input weights effectively or simply, how about to control the hidden layer's outputs? That to say, we can only select some hidden layer's outputs that relate the prediction variables more closely to calculate output weights using pseudo inverse method.

Motivated by the above problems, a modified randomized weight neural networks based on MI is proposed in this paper. At first, the input variables and the random chosen input weights feed into certain activation function to produce outputs of the hidden layer. Then, MI values between these hidden layer's output and predicted variables are calculated, and these outputs with MI values higher than a preset threshold are selected. At last, pseudo inverse method is used to compute weights between these selected hidden layer's outputs and predicted variable. Therefore, input weights' randomization is controlled in some degrees. Simulation based on concrete compressive strength benchmark dataset is used to validate the proposed method.

2. Randomized Weights Neural Networks

Suppose that SLFNs with L hidden nodes can be represented as:

$$f(x) = \sum_{i=1}^{L} \beta_i G_i(\boldsymbol{a}_i, b_i, \boldsymbol{x}) = \boldsymbol{\beta} \cdot \boldsymbol{h}(\boldsymbol{x}) \tag{1}$$

where,

$$g_i = G_i(\boldsymbol{a}_i, b_i, \boldsymbol{x}) = g(\boldsymbol{a}_i \cdot \boldsymbol{x} + b_i) \tag{2}$$

g_i denotes the activation function of the ith hidden node, $\boldsymbol{a}_i$ is the input weights connecting the input layer to the ith hidden node, b_i is the bias of the ith hidden node, β_i is the output weight connecting the ith hidden node to the output layer, and $\boldsymbol{h}(\boldsymbol{x})$ is the mapping output of the hidden layer, can be denoted as

$$\boldsymbol{h}(x) = [G_1(\boldsymbol{a}_1, b_1, x), \cdots, G_i(\boldsymbol{a}_i, b_i, x)] \tag{3}$$

Then, Equation (1) can be rewritten as:

$$\boldsymbol{H\beta} = \boldsymbol{Y} \tag{4}$$

where,

$$\boldsymbol{H} = [\boldsymbol{h}_1(\boldsymbol{x}), \cdots, \boldsymbol{h}_L(\boldsymbol{x})]_{k \times L} \tag{5}$$

$$\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_L,]^{\mathrm{T}}{}_{L \times 1} \tag{6}$$

$$\boldsymbol{Y} = [\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_K,]^{\mathrm{T}}{}_{k \times 1} \tag{7}$$

Theoretically, SLFNs are able to approximate any continuous target functions with enough hidden layer nodes using the randomized input weights. Give a training set $[\boldsymbol{x}_l, y_l], l \in [1, k]$, the randomized weight neural networks aim to reach the smallest training error and the smallest norm of output weights jointly.

$$\begin{aligned} &\text{minmize}: \quad \sum \|\beta_i \cdot h(x_i) - y_i\|^2 \\ &\text{and} \\ &\text{minmize}: \quad \|\boldsymbol{\beta}\| \end{aligned} \tag{8}$$

The solution can be analytically determined by the expression below:

$$\hat{\boldsymbol{\beta}} = \boldsymbol{H}^{+}\boldsymbol{Y} \tag{9}$$

where $\boldsymbol{H}^{+}$ is the Moore-Penrose generalized inverse of matrix $\boldsymbol{H}$.

The reason of using Moore-Penrose generalized inverse is that matrix $\boldsymbol{H}$ may be singular and/or be not square. The relations between $\boldsymbol{H}^{+}$ and $\boldsymbol{H}$ include: $\boldsymbol{H}\boldsymbol{H}^{+}\boldsymbol{H} = \boldsymbol{H}$, $\boldsymbol{H}^{+}\boldsymbol{H}\boldsymbol{H}^{+} = \boldsymbol{H}^{+}$, $(\boldsymbol{H}\boldsymbol{H}^{+})^{\mathrm{T}} = \boldsymbol{H}\boldsymbol{H}^{+}$ and $(\boldsymbol{H}^{+}\boldsymbol{H})^{\mathrm{T}} = \boldsymbol{H}^{+}\boldsymbol{H}$.

In particular, when $\boldsymbol{H}$ has full column rank,

$$\boldsymbol{H}^{+} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}} \tag{10}$$

And when $\boldsymbol{H}$ has full row rank,

$$\boldsymbol{H}^{+} = \boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}. \tag{11}$$

3. Mutual Information Based Feature Selection

3.1. Mutual Information

Information entropy can quantify the uncertainty of the random variables and scale the amount of information shared by these variables. Thus, it has been widely used in many fields. The entropy can be represented as:

$$\mathrm{H(X)} = -\sum \mathrm{p}(x)\log p(x) \tag{12}$$

where, $p(x)$ is the margin probability density.

Mutual information (MI) can measure the mutual dependence of two variables, which is defined as:,

$$MI(\boldsymbol{Y};\boldsymbol{X}) = \sum\sum p(y,x)\log\frac{p(x,y)}{p(x)p(y)} = H(\boldsymbol{Y}) - H(\boldsymbol{Y}\,|\,\boldsymbol{X}) \tag{13}$$

where, $p(y,x)$ is the joint probability density, and $H(\boldsymbol{Y}\,|\,\boldsymbol{X})$ is the conditional entropy at $\boldsymbol{X}$ is known, which is calculated as

$$H(\boldsymbol{Y}\,|\,\boldsymbol{X}) = -\sum\sum p(y\,|\,x)\log(p(y\,|\,x)) \tag{14}$$

For the continuous random variables,

$$\mathrm{H(X)} = -\int_x \mathrm{p}(x)\log p(x)dx \tag{15}$$

$$H(\boldsymbol{Y}\,|\,\boldsymbol{X}) = -\iint_{x,y} p(y,x)\log(p(y\,|\,x))dxdy \tag{16}$$

$$MI(\boldsymbol{Y};\boldsymbol{X}) = \iint_{x,y} p(y,x)\log\frac{p(x,y)}{p(x)p(y)}dxdy \tag{17}$$

3.2. Simple Feature Selection Based on Mutual Information

Mutual information feature select (MIFS) algorithm can be described as: calculate MI values between each input feature and output variable, then select the input features with the bigger MI values and penalize the others features have the bigger MI values with the selected features, and obtain the best input feature sub-set using the greedy search method [14]. This method is time-consuming for select features from high dimensional data.

A simple method based on MI is: 1) Calculate MI values between each input feature and output variables; 2) Given a pre-set threshold value of the MI based on prior knowledge; 3) The features with higher MI values than the threshold are selected. How to select the optimal pre-threshold value is an open question.

4. MI Based on Modified Randomized Weights Neural Networks

The proposed MI based modified randomized weights neural networks model are shown in **Figure 1**.

As shown in **Figure 1**, after obtain the mapping outputs of the hidden layer nodes $\boldsymbol{h}_1(\boldsymbol{x}),\cdots,\ \boldsymbol{h}_L(\boldsymbol{x})$, the MI values between these outputs and predicted variable are calculated with:

$$MI\left(\boldsymbol{y};\boldsymbol{h}_i\right)=\iint p(\boldsymbol{y},\boldsymbol{h}_i)\log\frac{p(\boldsymbol{y},\boldsymbol{h}_i)}{p(\boldsymbol{h}_i)p(\boldsymbol{y})}d(\boldsymbol{h}_i)d\boldsymbol{y} \tag{18}$$

Given that pre-set threshold value θ_{MI}, the following equation is used to select hidden layer's outputs:

$$\zeta_{\boldsymbol{h}_i}=\begin{cases}1 & \text{if} \quad Muin\left(\boldsymbol{y},\boldsymbol{h}_i\right)\geq\theta_{\mathrm{MI}}\\ 0 & \text{else} \quad Muin\left(\boldsymbol{y},\boldsymbol{h}_i\right)<\theta_{\mathrm{MI}}\end{cases} \tag{19}$$

We denote these hidden layer's outputs with $\zeta_{\boldsymbol{h}_i}=1$ as:

$$\boldsymbol{H}^{\mathrm{sel}}=[\boldsymbol{h}_1^{\mathrm{sel}}(\boldsymbol{x}),\cdots,\boldsymbol{h}_{L_{\mathrm{sel}}}^{\mathrm{sel}}(\boldsymbol{x})]_{k\times L_{\mathrm{sel}}} \tag{20}$$

where, L_{sel} is the number of the selected hidden layer's outputs.

Therefore, $\boldsymbol{H}^{\mathrm{sel}}$ has less randomization than that of the original $\boldsymbol{H}$. Output weights are also computed using the Moore-Penrose method with:

$$\hat{\boldsymbol{\beta}}=(\boldsymbol{H}^{\mathrm{sel}})^{+}\boldsymbol{Y} \tag{21}$$

Consideration problem of the learning parameters' selection, the MI based randomized weights algorithms can be represented as the following optimization problem:

$$\begin{aligned}&\text{Min} \quad E=\sqrt{\sum_{j=1}^{k}(\hat{y}_j-y_j)^2\Big/k}=\sqrt{\sum_{j=1}^{k}(f(L,\theta_{\mathrm{MI}},\boldsymbol{\beta})_j-y_j)^2\Big/k}\\ &s.t. \quad \begin{aligned}&L_{\min}\leq L\leq L_{\max}\\ &\theta_{\min}\leq\theta_{\mathrm{MI}}\leq\theta_{\max}\end{aligned}\end{aligned} \tag{22}$$

Some intelligent optimization methods can be used to address this problem.

5. Application on Modeling Concrete Compressive Strength

Concrete compressive strength data obtained by the experimental studies of the group led by I.C. Yeh in Taiwan

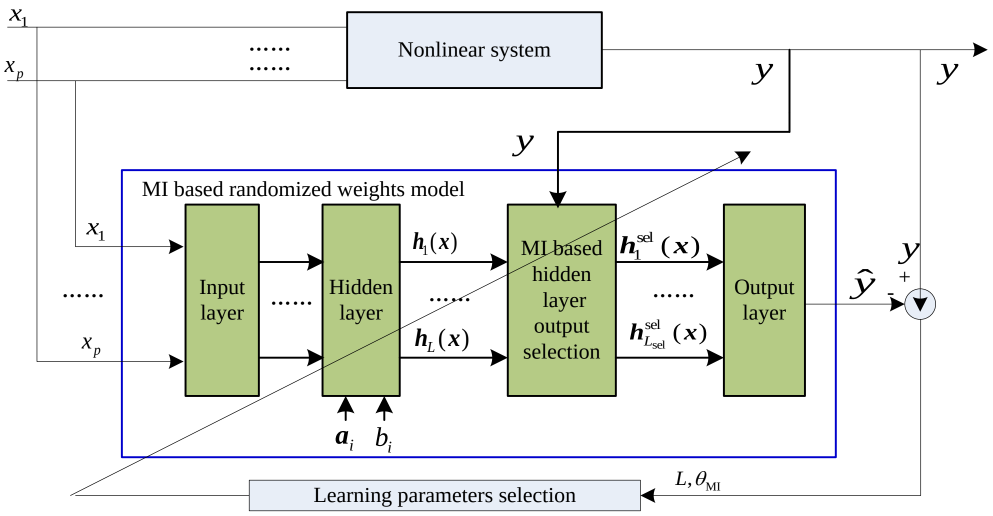
x_1
x_p
......
......
Nonlinear system
y
y
y
MI based randomized weights model
x_1
......
x_p
Input layer
......
Hidden layer
$\boldsymbol{h}_1(\boldsymbol{x})$
......
$\boldsymbol{h}_L(\boldsymbol{x})$
MI based hidden layer output selection
$\boldsymbol{h}_1^{\mathrm{sel}}(\boldsymbol{x})$
......
$\boldsymbol{h}_{L_{\mathrm{sel}}}^{\mathrm{sel}}(\boldsymbol{x})$
Output layer
$\hat{y}$
y
+
-
$\boldsymbol{a}_i$ b_i
Learning parameters selection
L,θ_{MI}

Figure 1. MI based modified randomized weights neural networks model.

Chung Hua University [15]. This dataset contains 1030 samples, each sample has nine columns. The first 7 columns are the input parameters, namely cement, blast furnace slag, fly ash, water, super plasticizer, coarse aggregate and fine aggregate in concrete per cubic content of the various ingredients of concrete placement. The eighth column is conserved days, and the last column is concrete compressive strength.

Given that L = 300, the MI values between hidden layer's outputs and predicted variable are shown in **Figure 2**.

Figure 2 shows that the maximum MI value is almost 10 times than that of the minimum value. Thus, the hidden layer's outputs are not stability. It is needed to select outputs with high MI values.

The original randomized weights algorithm and MI based modified version are compared with different hidden nodes' number and different MI pre-set threshold values. In order to overcome the randomization of the initial weights, the mean root mean square errors (MRMSEs) with repeated 100 times are used to estimate the model's prediction accuracy. Statistical results are shown in **Table 1**.

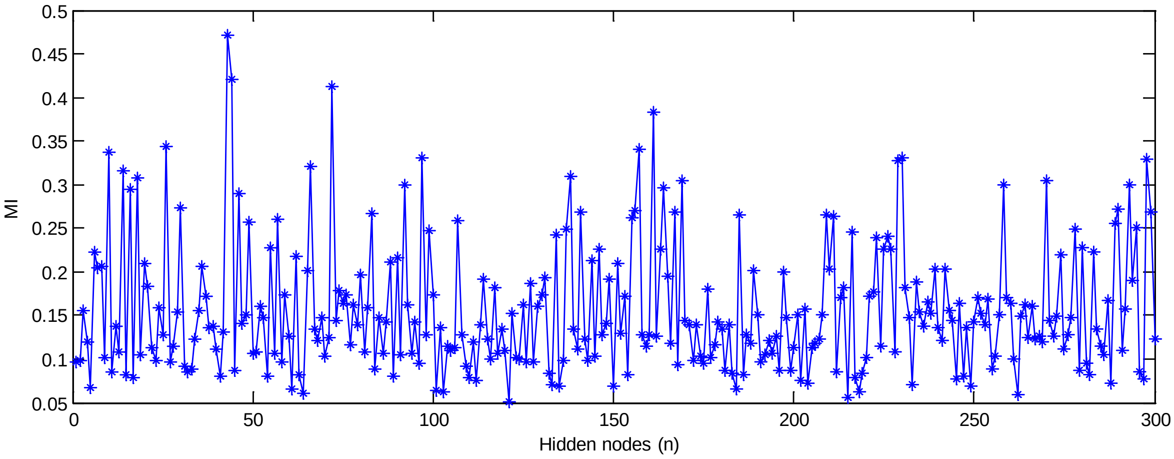

Figure 2. MI values between hidden layer's outputs and predicted variable.

Table 1. Statistical results (MRMSEs) of different learning parameters with repeated 100 times.

θ_{MI} L	Original method (MRMSEs)	MI based modified method (MRMSEs, L_{sel}) with different θ_{MI}							
		0.1* θ_{max}	0.2* θ_{max}	0.3* θ_{max}	0.4* θ_{max}	0.5* θ_{max}	0.6* θ_{max}	0.7* θ_{max}	θ_{max}
L = 10	12.37	(12.41, 10)	(**12.27, 9.8**)	(12.57, 9.1)	(13.97, 6.6)	(14.76, 4.9)	(16.42, 3.1)	--	0.2703
L = 20	10.36	(**10.25, 20**)	(10.31, 19.2)	(10.73, 15.4)	(11.73, 9.85)	(13.60, 6.03)	(15.04, 3.75)	--	0.3319
L = 30	9.713	(**9.675, 29.9**)	(9.702, 28.1)	(10.16, 21.3)	(10.95, 13.3)	(12.33, 8.07)	(14.77, 4.69)	--	0.3572
L = 40	9.518	(**9.444, 39.9**)	(9.621, 37.2)	(9.787, 27.2)	(10.61, 16.2)	(11.78, 9.11)	(13.89, 5.09)	--	0.3707
L = 50	9.755	(9.799, 49.9)	(9.638, 46.2)	(**9.591, 33.2**)	(10.21, 19.3)	(11.21, 11.6)	(13.19, 6.37)	--	0.3755
L = 60	10.17	(10.12, 59.8)	(9.772, 53.9)	**(9.486, 36.2)**	(10.16, 19.7)	(11.26, 10.8)	(13.02, 6.36)	--	0.4042
L = 70	10.38	(10.71, 69.8)	(10.14, 62.9)	**(9.453, 41.8)**	(9.973, 23.1)	(11.02, 12.83)	(12.14, 7.19)	--	0.4050
L = 80	11.18	(11.33, 79.8)	(10.62, 70.5)	**(9.625, 44.8)**	(9.738, 24.1)	(10.79, 13.0)	(12.68, 7.15)	--	0.4185
L = 90	12.48	(12.22, 89.7)	(11.24, 80.2)	**(9.634, 53.3)**	(9.736, 27.93)	(10.56, 15.28)	(11.50, 8.96)	(13.72, 4.82)	0.4113
L = 100	13.30	(13.06, 100)	(12.88, 99.0)	(12.20, 89.6)	(10.47, 70.5)	(9.885, 48.5)	(**9.614, 32.5**)	(10.00, 22.6)	0.4186
L = 200	535.1	(488.6, 199)	(47.26, 1.65)	(12.38, 94.0)	(**9.678**, 46.92)	(10.01, 23.87)	(10.95, 12.51)	(12.46, 5.99)	0.4612
L = 300	167.5	(166.4, 2.98)	(253.1, 240)	(20.24, 128)	(10.69, 15.53)	(**9.646, 31.5**)	(10.51, 15.4)	(12.05, 7.33)	0.4798
L = 400	132.2	(135.3, 397)	(156.7, 314)	(58.05, 167)	(11.66, 81.5)	(**9.874, 41.2**)	(10.24, 19.9)	(11.26, 9.53)	0.4854
L = 500	121.4	(118.1, 496)	(379.7, 382)	(512.9, 197)	(12.56, 93.6)	(**9.957, 45.1**)	(10.16, 21.6)	(10.95, 11.0)	0.5070

Table 1 shows that: 1) The maximum MI values based on different learning parameters between hidden layer's output and predicted variable increase with the number of the hidden nodes; 2) All smallest prediction errors with different learning parameters (L, θ_{MI}) occur with L = 30 - 40; Thus, it may be the best range for this benchmark dataset; 3) The biggest prediction errors occur at about L = 200. The reason may be relate to the Moore-Penrose method; 4) The prediction performance isn't much improved with the modified approach with L = 40. However, with the other L values, the prediction performance can be improved much with suitable MI pre-set threshold value. Therefore, the largest prediction error problems at L = 200 can be avoided with the MI based modified approach. Thus, the proposed method has better robustness than that of the original randomized weighting algorithm.

6. Conclusion

This paper proposes new mutual information based randomized weights neural networks. Input weights of the hidden layer are produced randomly as normal randomized algorithm. Not all the outputs of the hidden layer are used to compute output weights. Mutual information based simple feature selection method is used to select hidden layer's outputs. These selected outputs are used to compute weights of hidden layer with pseudo inverse method. Concrete compressive strength benchmark dataset is used to validate this method. More researches will address some theoretically analysis and to validate this idea with more benchmark datasets.

Acknowledgements

The research was sponsored by the post doctoral National Natural Science Foundation of China (2013M532118, 2015T81082), National Natural Science Foundation of China (61573364, 61273177), State Key Laboratory of Synthetical Automation for Process Industries, China National 863 Projects (2015AA043802).

References

[1] Shang, C., Yang, F., Huang, D.X. and Lu, W.X. (2014) Data-Driven Soft Sensor Development Based on Deep Learning. *Journal of Process Control*, **24**, 223-233. http://dx.doi.org/10.1016/j.jprocont.2014.01.012

[2] Pao, Y.H. and Takefuji, Y. (1992) Functional-Link Net Computing, Theory, System Architecture, and Functionalities. *IEEE Computer*, **25**, 76-79. http://dx.doi.org/10.1109/2.144401

[3] Igelnik, B. and Pao, Y.H. (1995) Stochastic Choice of Basis Functions in Adaptive Function Approximation and the Functional-Link Net. *IEEE Trans. Neural Network*, **6**, 1320-1329. http://dx.doi.org/10.1109/72.471375

[4] Huang, G.B., Chen, L. and Siew, C.K. (2006) Universal Approximation Using Incremental Constructive Feedforward Networks with Random Hidden Nodes. *IEEE Transactions on Neural Networks*, **17**, 879-892. http://dx.doi.org/10.1109/TNN.2006.875977

[5] Tapson, J. and Schaik, A.V. (2013) Learning the Pseudoinverse Solution to Network Weights. *Neural Networks*, **45**, 94-100. http://dx.doi.org/10.1016/j.neunet.2013.02.008

[6] Alhamdoosh, M. and Wang, D.H. (2014) Fast Decorrelated Neural Network Ensembles with Random Weights. *Information Sciences*, **264**, 104-117. http://dx.doi.org/10.1016/j.ins.2013.12.016

[7] Bartlett, P.L. (1997) For Valid Generalization, the Size of the Weights Is More Important Than the Size of the Network. *IEEE Conference on Neural Information Processing Systems*, MIT Press, Cambridge, 134-140.

[8] Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H. (2007) Greedy Layer-Wise Training of Deep Networks. *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, 153-160.

[9] de la Rosa, E. and Yu, W. (2015) Nonlinear System Identification Using Deep Learning and Randomized Algorithms. *IEEE International Conference on Information and Automation (ICIA*2015), Lijing, 274-279. http://dx.doi.org/10.1109/ICInfA.2015.7279298

[10] Liu, H.W., Sun, J.G., Liu, L. and Zhang, H.J. (2009) Feature Selection with Dynamic Mutual Information. *Pattern Recognition*, **42**, 1330-1339. http://dx.doi.org/10.1016/j.patcog.2008.10.028

[11] Peng, H.C., Long, F.H. and Ding, C. (2005) Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 1226-1238. http://dx.doi.org/10.1109/TPAMI.2005.159

[12] Tan, C. and Li, M.L. (2008) Mutual Information-Induced Interval Selection Combined with Kernel Partial Least Squares for Near-Infrared Spectral Calibration. *Spectrochimica Acta Part A*: *Molecular and Biomolecular Spectroscopy*, **71**, 1266-1273. http://dx.doi.org/10.1016/j.saa.2008.03.033

[13] Tang, J., Chai, T.Y., Yu, W. and Zhao, L.J. (2012) Feature Extraction and Selection Based on Vibration Spectrum with Application to Estimate the Load Parameters of Ball Mill in Grinding Process. *Control Engineering Practice*, **20**, 991-1004. http://dx.doi.org/10.1016/j.conengprac.2012.03.020

[14] Battiti, R. (1994) Using Mutual Information for Selecting Features in Supervised Neural Net Learning. *IEEE Transaction on Neural Network*, **5**, 537-550. http://dx.doi.org/10.1109/72.298224

[15] Yeh, I.C. (1998) Modeling of Strength of High Performance Concrete Using Artificial Neural Networks. *Cement and Concrete Research*, **28**, 1797-1808. http://dx.doi.org/10.1016/S0008-8846(98)00165-3