An Effective Intelligent Self-Construction Multilayer Perceptron Neural Network

Amany S. Saber,
Moustafa S. El-Koliel
Nuclear Research Center,
Atomic Energy Authority,
Cairo, Egypt

Mohamed A. El-Rashidy,
Taha E.Taha
Faculty of Electronic Engineering,
Menoufiya University,
Cairo, Egypt

ABSTRACT

A new classifier algorithm based on Multilayer Perceptron Neural Network (MPNN), Apriori association rules, and Particle Swarm Optimization (PSO) models is proposed. It provides a comprehensive analytic method for establishing an Artificial Neural Network (ANN) with self-organizing architecture by finding an optimal number of hidden layers and their neurons, less number of effective features of data set, and better topology for internal connections. The performance of the proposed algorithm is evaluated using a number of benchmark data sets including Breast Cancer, Iris, and Yeast. Experimental results demonstrate the effectiveness and the notability of the proposed algorithm comparing with recently existed ANN learning and classification algorithms.

General Terms

Data Mining, Artificial Neural Networks.

Keywords

Classification, Artificial neural network, Apriori association rules, Particle swarm optimization.

1. INTRODUCTION

Artificial Neural Networks (ANNs) are one of the most effective, flexible, powerful and attractive techniques used for classification tasks. ANNs have many other names such as, neural nets, neurocomputers, and parallel distributed processing systems. They are composed of large number of processing units called neurons and connections between the units. Neurons have weighted inputs, threshold values, activation function, and an output. ANNs have been used in different applications, such as mathematical modeling, classification, clustering, forecasting, image and face recognition, medicine, industry, and control. There are many characteristics or properties of artificial neural networks that make them better for classification than traditional methods: (1) Neural networks can learn and identify their function based only on the given training data, (2) Neural networks can learn in existence of noise, (3) They have the ability to generalize, where they may provide outputs with lack of data, (4) They have dynamic infrastructure. The Feed Forward Neural Networks (FNNs) are the simplest form of ANN, and they are one of the most famous neural network topology that has been widely used in classifications. There are many different kinds of learning algorithms used by neural networks for training FNN. There are many problems that have to be discussed when using a FNN. But the most important problem is to determine the optimal architecture of network (number of hidden layers and their neurons, topology connection, and adaptive strategy planning) before the training process to achieve the desired performance. There are a number of different algorithms that have been introduced which attempt to solve this problem. These algorithms are modifying the connection weights and also the network architecture as training goes. These algorithms can be classified into two main classes. The first class includes algorithms that start with an oversized network and then gradually prunes the redundant nodes or connections in order to enhance performance. These algorithms are called pruning algorithms. There are two main subcategories of pruning methods: (i) pruning based on modifying the error function and (ii) pruning based on sensitivity measures. Many different pruning methods have emerged in [13] [15][17].

The second class includes algorithms that start with a minimal network and gradually insert nodes or connections during training when it is needed to enhance the learning of the network. These methods are called constructive algorithms. The most common constructing algorithms are growing cell structure (GCS) [4], constructive back-propagation (CBP) [14], adaptively constructing multilayer FNN [16], extreme learning machine with adaptive growth of hidden nodes (AG-ELM) [23], enhanced incremental extreme learning machine (EI-ELM) [8], and etc. A hybrid approach which merges the constructive and pruning algorithms is also introduced. This approach performs a constructive phase first, and then a pruning phase. The hybrid approach algorithms are trying to enhance the execution performance of the previous two algorithms by employing a pruning phase coupling with a constructive phase. A pruning approach is applied to enhance the selection, and remove the redundant or unnecessary hidden neurons (or connections). Then, the architecture of FNNs can be designed automatically. A number of other improved constructive and pruning algorithms can be found in[2] [9] [10].

In this paper a new proposed algorithm is presented to find the optimal network architecture for simple and complex data sets. The proposed algorithm is based on the constructive and pruning concepts. This was carried out through functioning design process that includes new strategy for learning algorithm to select effective features of the data sets, find an optimal number of hidden layers and their neurons, and topology connections. The proposed algorithm can reach to a consolidated structure size, which it based on a priori knowledge from many training phases, assisted by Apriori algorithm of association rules, and it used Particle swarm algorithm for an optimization process to minimize the number of training phases. The structure of this paper is as follows. In section 2, the related literature for Artificial Neural Networks, Apriori, and Particle Swarm Optimization algorithms are explained. In section 3, a description of the proposed algorithm will be detailed. In section 4, the extensive experimental results of proposed algorithm on different data sets will be discussed. In the last section, the concluding remarks will be offered.

2. RELATED WORK

2.1 Artificial Neural Network

Artificial Neural Networks (ANNs) are a tool used for solving problems by emulating the connection between neurons in the nervous system of the human brain. ANNs are arranged in layers where these layers are consists of large number of artificial units connected together into a neural network called “neurons”, and weighted connections between the units. Neurons have weighted inputs, threshold values, activation function, and an output. Neurons communicate with each other via weighted connections to perform the desired function. Each neuron within the network is a simple processing unit which simply adds together one or more inputs from other neurons, changes its activation, and produces an output. The activation function for each neuron is usually a sigmoid function in the back-propagation architecture, and a Gaussian function in the radial basis architecture. Fig. 1 shows the neurons connections in the neural network [6].

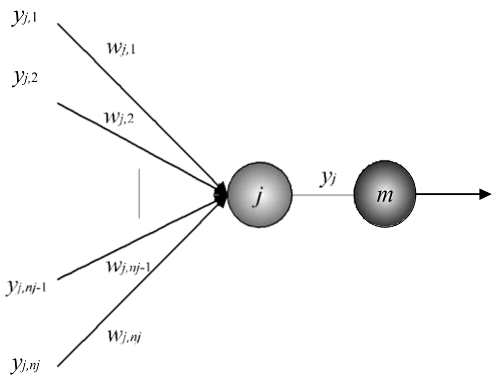


Fig 1: Connection of a neuron j with the other neurons.

In fig. 1, output node of neuron *j* is calculated as:

$$y_j = f_j(net_j), \qquad (1)$$

where, y_j is the output of neuron *j*, $f_j(x)$ is the activation function of neuron *j*, and net_j is the weighted sum neurons of neuron *j* and is calculated as:

$$net_j = \sum_{i=1}^{n_j} w_{j,i}\, y_{j,i} \ , \qquad (2)$$

where, $w_{j,i}$ is the weight between the i^{th} input neuron and neuron *j*, $y_{j,i}$ is the i^{th} input of neuron *j*, and n_j is the total number of input neurons of neuron *j*. There are many factors affect the performance of ANNs, such as the activation functions, size of training samples, network topology, learning algorithm. A schematic diagram of an ANN is shown in fig. 2, where the circles are artificial neurons and the connections represent weights. Three types of layers are shown input, hidden, and output layers.

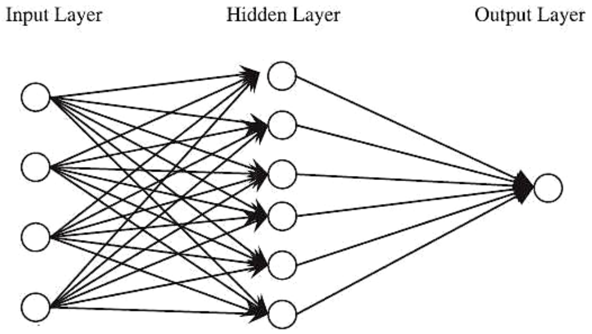


Fig 2: Schematic diagram of an ANN.

Back-propagation algorithm is the most familiar, powerful, and effective algorithm used to train the multilayer perception (MLP) networks. It consists of an input layer, an output layer, and at least one hidden layer. Each layer is fully connected to the succeeding layer. Learning happens in ANN during training phase in order to adjust the weights of the interconnections between neurons. In ANN learning process; weights and ANN architecture are initialized randomly, and then the training starts, training samples are applied to the input layer of the network, and desired outputs are compared at the output layer of the network. The difference between the output layer and the desired output is back-propagated to the previous layers. Mean square error E [19] is defined to evaluate the training process, it is calculated as:

$$E = 1/2 \sum_{i=1}^{p} \| \mathrm{o}_i - \mathrm{t}_i \|^2 \ , \qquad (3)$$

where, o_i and t_i denote the output and the target output of the network respectively for all training samples. The weights in the network are updated to let the quadratic error *E* as minimum as possible. Error *E* is minimized by using an iterative process of gradient descent [19],

$$\Delta w_{k+1} = \alpha_k(-\mathrm{g}_k), \qquad (4)$$

$$w_{k+1} = w_k + \Delta w_{k+1}, \qquad (5)$$

where, w_k is a vector of current weights, g_k is the current gradient, and α_k is the learning rate.

2.2 Apriori Algorithm

Association rule analysis is a common and effective method for finding all interesting co-occurrence relationships between data items in large databases. Its objective is to determine strong rules between variables discovered in databases using various measures. Association rules were first introduced by [1] for finding out regularities between products in supermarkets. Association rule analysis has been used in different applications, such as web mining, scientific data analysis, bioinformatics, and medical diagnosis. An association rule is an implication of the form X $\rightarrow$ Y, where X,Y $\subset$ data set attributes, and X $\cap$ Y = $\varnothing$. The itemsets X and Y are called antecedent and consequent of the rule respectively. Support and confidence are used to measure the strength of the rule. The support of a rule, X $\rightarrow$ Y, is the percentage of transactions in T that contains X $\cup$ Y, while the confidence is the percentage of transactions in T that contain X also contain Y.

The process of discovering all association rules can be divided into two steps [1]. The first step is to find all frequent Itemsets. A frequent itemset is an itemset that has transaction support above minimum support. In second step, use the frequent itemsets to find the confident association rules. A confident association rule is a rule with confidence above minimum confidence. Whatever the algorithm used in discovering all association rules, it should find the same set of rules. There is a large number of algorithms but the Apriori algorithm introduced in [1] is the most familiar and effective mining algorithm.

2.3 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is first introduced by [12] is a computational optimization method used for solving optimization problems. PSO is based on a swarm intelligence, which it comes from social motion manner of bird and fish flocking. In PSO, particles are a simple program proxies moving, live, and working together to find an optimum solution for optimization problem. PSO is initialized with a group of random particles; the location of each particle stands for a candidate solution for the optimization problem. Each particle is seeking for optimal locations in the search space by modifying its speed. The location of each particle is influenced by the most optimistic location during its movement in the search space and the most optimistic particle location in its surrounding.

Each particle is considered as a point in a D-dimensional space. $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ is the location of i^{th} particle in D-dimension. The speed for particle i is represented as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. Each particle is modified by two "best" values, the first one is the best location *pbest* found by this particle. The second value is the best location *gbest* gained till now by any particle in the surrounding. After getting the two best values, the particle modifying its speed and locations according to the following two equations [20]:

$$V_{id} = V_{id} + c_1 * rand() * (Pbest_{id} - X_{id}) + c2 * Rand() * (gbest_d - X_{id}), \qquad (6)$$

$$X_{id} = X_{id} + V_{id}\,, \qquad (7)$$

where, *c1* and *c2* are two positive learning factors, *rand()* and *Rand()* represents two random functions in the range [0,1]. $Pbest_{id}$ is the number d component of the best position particle i has ever found; $gbest_d$ is the number d component of the best position the surrounding particles have ever found.

3. PROPOSED ALGORITHM

We have two externally supplied inputs; the first input is the given training data set which is denoted by T, and the second input is empty data set denoted by D. The inputs to proposed algorithm can help to formalize the requirements and restrictions that architecture must fit. The quantities produced after applying the proposed algorithm are the outputs which are organized as follows: the number of hidden layers of neural network architecture is denoted by J, the number of nodes for each hidden layer J is denoted by I, the term TOP is used to represent neural network topology, the term RMSE is used to measure the root mean square error of the network, and finally the overall classification accuracy of the network is denoted by ACC.

The aim of this algorithm is to establish an optimal artificial neural network with self-organizing architecture by changing input features, number of hidden layers and their hidden neurons, network topology. It can reach a compact structure size, based on a priori knowledge from a set of training samples, assisted by Apriori algorithm for association rules and Particle swarm for optimization process. The proposed algorithm is organized in 16 steps as shown in fig. 3.

```
Input: Training data set T, and empty data set D.
Output: J, I, TOP, RMSE, and ACC number of hidden layers, number of nodes for hidden layer J, neural
        network topology, root mean squared error, and overall accuracy, respectively.
Step 1: Initialize original data: by normalization process to form training data set.
Step 2: Initialize neural network architecture: included $H_J$, and $N_{Ji}$ to set J=0, and i=0.
Step 3: Update D: to equal T.
Step 4: Set iterative variable: J = J +1.
Step 5: Data preparation: D is represented as nominal attributes.
Step 6: Particle initialization and PSO parameters setting: including number of particles (n), particle
        dimension (m), number of maximal iterations ($k_{max}$), error limitation of the fitness function,
        velocity limitation ($V_{max}$), and inertia weight for particle velocity (w).
Step 7: Set iterative variable: k = k + 1.
Step 8: Apply Apriori algorithm: using D to find related sets of its attributes
Step 9: Update i and topology of $N_{Ji}$.
Step 10: If i=1 go to step 16.
Step 11: Training neural network: using T.
Step 12: Compute the fitness function value of each particle.
Step 13: Stop condition checking: if intra stopping criteria are met go to step 15, if inter stopping criteria are
         met go to step 16.
Step 14: Update the particle position, go to step 7.
Step 15: Update D: to equal output of $H_J$, go to step4.
Step16: Compute RMSE and overall accuracy: by training neural network using T.
```

Fig 3: Pseudo code of proposed algorithm.

First normalization process is applied to reduce redundancies of data in database, and manipulate attributes in a desired appropriate form. An initial Neural Network is then created consisting of: input layer, H_J (N_{Ji} neurons), and output layer, network topology is not recognized yet. Specifying minimum support and minimum confidence; Apriori algorithm finds the most effective sets of related data set inputs (features). Each extracted features set S_i construct a hidden node N_{Ji}, where features of this set S_i are input of the new constructed hidden node N_{Ji}. So, all possible hidden nodes are equal to discovered sets Si and their features are input for constructed nodes to build network topology. Particle swarm optimization (PSO) algorithm is used to find an optimum number of hidden nodes in each hidden layer which achieves highest accuracy. A dynamic strategy is used to train neural network; this strategy can insert or remove hidden nodes and layers, and it can changes topology of neural network via the measurement of classification accuracy in the training process.

Classification accuracy and RMSE are used to decide whether neurons should be inserted or deleted. Initially, a network with one hidden layer and one hidden neuron is constructed, and then new hidden neurons (only one neuron is inserted at a time) are incrementally changed. Each new hidden neuron and hidden layer is trained to minimize the current network error in order to enhance the performance of the network. A pruning approach is applied to enhance the selection, and remove the unnecessary hidden neurons (or connections) either during training or after convergence to a local minimum. Finally we stop when pre-specified error requirement is found, or no performance refinement is noticed.

4. EXPERIMENTAL RESULTS

Experimental studies are presented to prove the effectiveness and the notability of proposed algorithm. We applied our proposed method to three kinds of benchmark data sets provided by the UCI Repository of Machine Learning [22], namely, Breast Cancer, Iris, and Yeast. For each dataset, we normalized the data over each feature value. Table 1 shows the total number of instances in each data set, input features, and classes.

Table 1. Characteristics of UCI benchmarks and a qualitative analytical chemistry data set.

Data Set	Patterns	Input features	Classes
Breast Cancer	699	9	2
Iris	300	4	3
Yeast	1484	8	10

First experiment, the altitude of diagnostic accuracy using proposed model can be appeared by sensitive rates, which compared with Constructing and Pruning Neural Network (CP-NN) [5], Extended Fourier Amplitude Sensitivity Test (EFAST) [13], Adaptive Growing and Pruning Neural Network Control (AGPNNC) [7], Adaptive Merging and Growing Algorithm (AMGA) [10], Memetic Pareto artificial neural networks (MPANN-HN) [3], and Enhanced Incremental Extreme Learning Machine (EI-ELM) [8] algorithms.

The reported results in table 2 are averaged over 50 trials of simulations. The averaged values of the following parameters were used to measure the performance: the mean accumulated CPU time the training process took, the number of preserved nodes in the hidden layers, and the test recognition rate.

Table 2. Performance comparison of the different ANN learning algorithms

Data Set	Algorithms	Mean CPU time (s)	No. of hidden nodes	Recognition rate (%)
Breast Cancer	*Proposed algorithm*	77.02	6	*97.2818*
	CP-NN	13.26	21	68.94
	EFAST	-	2	-
	AGPNNC	27.83	32	64.48
	AMGA	21.29	28	65.88
	MPANN-HN	-	2	-
	EI-ELM	0.76	121	67.24
Iris	*Proposed algorithm*	132.49	10(3/7)	*97.3333*
	CP-NN	3.03	21	96.32
	EFAST	-	2	-
	AGPNNC	5.18	23	95.29
	AMGA	5.01	23	95.53
	MPANN-HN	-	2	-
	EI-ELM	0.11	112	97.32
Yeast	*Proposed algorithm*	284.45	17	*63.0054*
	CP-NN	88.02	43	53.26
	EFAST	-	2	-
	AGPNNC	189.21	62	53.17
	AMGA	178.23	57	51.28
	MPANN-HN	-	2	-
	EI-ELM	2.54	156	53.17

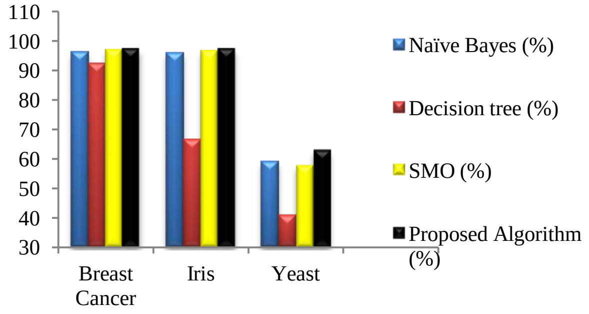
The highest achieved results are same as the original papers.

Table 3. Comparison of hidden layer numbers and selected features

Data Set	Algorithms	No. of hidden nodes			Selected Features
		H1	H2	H3	
Breast Cancer	*Proposed algorithm*	*6*	-	-	*6*
	AGPNNC	32	-	-	9
	AMGA	28	-	-	9
	EI-ELM	121	-	-	9
Iris	*Proposed algorithm*	*3*	*7*	-	*2*
	AGPNNC	23	-	-	4
	AMGA	23	-	-	4
	EI-ELM	112	-	-	4
Yeast	*Proposed algorithm*	*17*	-	-	*8*
	AGPNNC	62	-	-	8
	AMGA	57	-	-	8
	EI-ELM	156	-	-	8

The results are same as the original papers.

From the results in table 2 and 3, it is clearly seen that, the proposed algorithm achieves the highest classification accuracy with lowest number of nodes in hidden layers, which leads to provide the efficiency of treatment service with as little as possible in the decision making time after the construction of neural network topology. However, the training time is significantly longer than the other self-organizing neural network algorithms by utilizing our proposed algorithm. Second experiment covered a comparison of acquired accuracy for using proposed method with the other classifiers including Naïve Bayes [11], Decision tree [21], and the sequential minimal optimization SMO [18]. The reported values in fig. 4 show that the proposed algorithm has good performance and performs well in comparison with other algorithms. By analyzing first and second experiment, proposed algorithm can self-organize the network structure by the research objects. The final structure of the proposed algorithm is the most precise comparing with the other self-organizing neural network and classifier algorithms. The final network structure is the most stable over the 50 trials of simulations. Besides, it is able to extract the most effective features of data sets that shown in table 3 over 50 independent runs in different data sets, which make the proposed algorithm faster than other approaches after reaching network structure by training phase.



	Breast Cancer	Iris	Yeast
Naïve Bayes (%)	96.1373	96	59.1644
Decision tree (%)	92.4177	66.6667	40.7008
SMO (%)	96.9957	96.6667	57.7493
Proposed algorithm (%)	*97.2818*	*97.3333*	*63.0054*

Fig 4: Comparison of average classification accuracy rates

5. CONCLUSION

In this paper, we have proposed a new algorithm as an alternative approach for pattern classification of different statistical dataset. The proposed algorithm can reach a consolidated structure size of artificial neural network with extracting most effective features, based on a priori knowledge from a set of training samples, assisted by Apriori algorithm for association rules and particle swarm for optimization process. Three well-known real life benchmark data sets, namely, Breast Cancer, Iris, and Yeast are used in order to show the suitability and effectiveness of the proposed model for classification tasks. Simulation results show that the proposed algorithm achieves the highest classification accuracy with lowest number of nodes in hidden layers in comparison to all alternative classifier algorithms, besides, it is able to extract the most effective features.

6. REFERENCES

[1] Agrawal, R., Imieliski, T., and Swami, A.1993.Mining association rules between sets of items in large databases. In Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD-1993).

[2] Bortman, M. and Aladjem, M.2009.A growing and pruning method for radial basis function networks. IEEE Trans. Neural Networks. 20(6),1039–1045.

[3] Caballero, J. C. F., Martinez, F. J., Hervas, C., and Gutierrez, P. A. 2010.Sensitivity versus accuracy in multiclass problems using memeticpareto evolutionary neural networks. IEEE Trans. Neural Networks. 21(5), 751–770.

[4] Fritzke, B. 1994. Growing cell structures a self-organizing network for unsupervised and supervised learning. Neural Networks. 7(9),1441–1460.

[5] Han, H. G. and Qiao, J. F.2013. A structure optimization algorithm for feed-forward neural network construction. Neurocomputing. 99, 347–357.

[6] Han, H. G., Wang, L. D., and Qiao, J. F.2013.Efficient self-organizing multilayer neural network for nonlinear system modeling. Neural Networks. 43, 22–32.

[7] Hsu, C. F. 2008. Adaptive growing-and-pruning neural network control for a linear piezo electric ceramic motor. Eng. Appl. Artif. Intell. 21(8), 1153–1163.

[8] Huang, G. and Chen, L. 2008. Enhanced random search based incremental extreme learning machine. Neurocomputing. 71, 3460–3468.

[9] Huang, D. S. and Du, J. X. 2008. A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. IEEE Trans. Neural Networks. 19(12), 2099–2115.

[10] Islam, M. M., Sattar, M. A., Amin, M. F., Yao, X., and Murase, K. 2009. A new adaptive merging and growing algorithm for designing artificial neural networks. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. 39(3), 705–722.

[11] John, G. H. and Langley, P. 1995. Estimating continuous distributions in baysian classifiers. In Proceedings of the 11th conference on uncertainty in artificial intelligence. Montreal, Canada, 338–345.

[12] Kennedy, J. and Eberhart, R. 1995. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, 1942-1948.

[13] Lauret, P., Fock, E. and Mara, T. A. 2006. A node pruning algorithm based on a Fourier amplitude sensitivity test method. IEEE Trans. Neural Networks. 17 (2), 273–293.

[14] Lehtokangas, M. 1999. Modelling with constructive back-propagation. Neural Networks. 12(4–5), 707–716.

[15] Leung, C. S., Wong, K. W., Sum, P. F. and Chan, L. W.2001. A pruning method for the recursive least squared algorithm. Neural Networks. 14(2), 147–174.

[16] Ma, L. and Khorasani, K. 2003.A new strategy for adaptively constructing multilayer feed-forward neural networks. Neurocomputing. 51, 361–385.

[17] Nielsen, A. B. and Hansen, L. K. 2008. Structure learning by pruning in independent component analysis. Neurocomputing. 71(10–12),2281–2290.

[18] Platt, J. 1999. Using analytic QP and sparseness to speed training of support vector machines. In Proceedings of the 13th annual conference on neural information processing systems. Denver, Colorado, 557–563.

[19] Rojas, R. 1996. Neural Networks: A Systematic Introduction. Berlin, Springer-Verlag.

[20] Shi, Y. and Eberhart, R. 1998. A modified particle swam optimizer. IEEE Word Congress on Computational Intelligence. 69-73.

[21] Sun, J. 2010. Application of Data Mining for Decision Tree Model of Multi-variety Discrete Production and Manufacture. Intelligent Information Technology and Security Informatics (IITSI). Jinggangshan, 724 – 728.

[22] UCI Machine Learning Repository websitehttp://archive.ics.uci.edu:80/ml/datasets.html.

[23] Zhang, R., Lan, Y., Huang, G. B., and Xu, Z. B. 2012. Universal approximation of extreme learning machine with adaptive growth of hidden nodes. IEEE Trans. Neural Networks Learn. Syst.. 23(2), 365–371.