

Analysis of Randomized Performance of Bias Parameters and Activation Function of Extreme Learning Machine

Prafull Pandey
M. Tech. Scholar

Dept. of Computer Science and Technology
UIT, Allahabad, India

Ram Govind Singh
Asst. Professor

Dept. of Computer Science and Technology
UIT, Allahabad, India

ABSTRACT

In Artificial Intelligence classification is a process of identifying classes of a different entities on the basis information provided from the dataset. Extreme Learning Machine (ELM) is one of the efficient classifiers. ELM is formed by interconnected layers. Each layer has many nodes (neurons). The input layer communicates with hidden layer with random weight and produces output layer with the help of activation function (transfer function). Activation functions are non-linear functions and different activation functions may produce different output on same dataset. Not every activation function is suited for every type classification problem. This paper shows the variation of average test accuracy with various activation functions. Along with it also has been shown that how much performance varied due to selection of random bias parameter between input and hidden layer of ELM.

General Terms

Classification, dataset, testing, training, imbalance, classifier, layers, neurons.

Keywords

Extreme machine learning, feedforward network, neural network, classification

1. INTRODUCTION

In previous years, several researches have been done on the classification problem and many algorithms have been proposed over it. Some most popular approaches are Multilayer Perceptron and Bayesian decision theory [10, 12], support vector machine [9] and some variants [8] of it. Artificial Neural Network is one of good example of classification technique which works on back-propagation method [11] and weight is adjusted by trial and error methods but it suffers from problem like stuck in local minima [2], overfitting problem [7] and large training time [5]. Support vector machine [9] is another popular classifier based on statistical learning theory and having better generalization precision and nonlinear data processing ability. Support Vector Machine (SVM) suffers from problem like choice of kernel function, high algorithm complexity in training and testing phase and extensive memory requirements [21]. Along with it other classifiers are Random Forest [18], Naive Bayes classifier [17], Logistic Regression [19] etc.

Recently Hung et.al [4] has proposed a least square based learning algorithm for classification and regression problem named Extreme Learning Machine. It is a single hidden layer feed forward neural network in which the parameters between input and hidden layer are assigned randomly and the

parameter between hidden layer and output layer is obtained analytically by using a mathematical concept, Moore-Penrose (MP) generalized inverse. ELM is now widely used in various fields [15] like face detection, pattern recognition, digital image processing, signal processing etc. The reason of selecting ELM is that, it is much simpler than other traditional feed forward neural network and it does not suffer from the problem like learning speed, learning epochs and training of parameters. So, it can overcome the problem of traditional classifier such as overfitting, stopping criteria etc.

The training of ELM using data set is very fast the weight between input and hidden layer are initialized randomly. The performance of ELM may suffer from randomization. An activation function is also used at hidden layer to approximate the output from input layer. The performance of activation also fluctuates from the selection of activation function because mapping of output depends on which activation function has been used. Several author has been analyzed the performance on the basis of various activation function like sigmoid [2], sign, hard limiter, triangular bias [20] and radial bias [17] function.

In this paper first it has been analyzed how much performance fluctuated due to randomization and variation in performance due to selection of different activation using ELM based classifier. Along with it, classifier performance is also observed using different number of hidden neurons. The performance results are presented on the factors of varying hidden node with different activation function.

The rest of paper is divided into 4 sections. Section no. 2 describes the working of ELM. Section no. 3 contains brief information about activation functions used for analyzing the performance of ELM. Section no. 4 contains experimental after simulation ELM on different datasets. Section 5 discuss about conclusion obtained from experimental result.

2. EXTREME LEARNING MACHINE

Extreme learning machine is one of the classifiers based on single hidden layer feed forward neural network structure (SLNF) [2, 315, 16]. It is an emerging learning technique provides efficient solutions for classification and regression problems. ELM can be used for both single and multi hidden layer. For analyzing and tuning up the classifiers, two random sets (imbalance sets) are created from dataset. One considered as training set and other as testing set. With the help of training set calculation of output weight is performed by the ELM. After calculating output weight classification is performed on the testing data [22]. In ELM, the output of network for i^{th} hidden node

$$O_k = f_i(x) = \sum \beta_j g_i(x_j), \quad (i)$$

where O_k represents output where k is number of classes, $\beta = [\beta_1, \dots, \beta_i]^T$ is the output weight matrix between the hidden nodes and the output nodes. $g_i(x_j)$ can also written as

$$g_i(x_j) = g(w_i x_j + b_i) \quad (ii)$$

where $g_i(x_j)$ is an activation function x_j is an input sample, b_i is a threshold of i^{th} hidden node and w_i is an input weight.

Given N training samples $\{(x_j, t_j)\}_{j=1}^N$, the ELM can resolve the following learning problem:

$$H \beta = T, \quad (iii)$$

where $T = [t^1, \dots, t^N]^T$ are target labels, and $H = [h^T(x_1), \dots, h^T(x^N)]^T$. The output weights β will be calculated from

$$\beta = H^\dagger T, \quad (iv)$$

where H^\dagger is the pseudo-inverse (Moore-Penrose generalized inverse) of matrix H . The equation (iii) can also be rewritten as

$$\beta = (H^T H)^{-1} H^T T \quad (v)$$

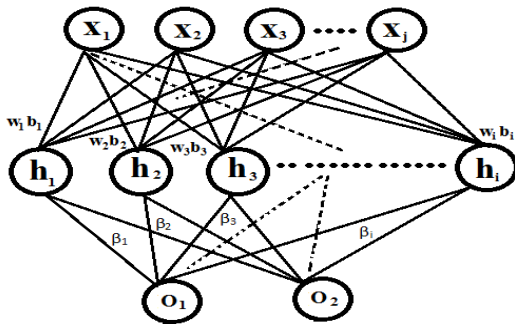


Fig 1: The structure of ELM with single hidden layer

3. ACTIVATION FUNCTIONS

The Activation function is used to calculate the output response of neuron. The neurons behavior is made up of two sections first is weighted sum of input and another is an activation function. The sum of the weighted input signal is applied with an activation function to obtain the response. For neurons of same layer, same activation function is used. There may be a linear as well as nonlinear activation functions. Linear function draw a straight line graph where as nonlinear functions draw a curved line [14]. The classifications problems are normally nonlinear in nature because rate of change of input and output is not in a constant ratio. The following are brief introductions about nonlinear activation functions used in ELM for classifications:

i) **Sine Function:** The sine function accepts real value and returns a real value between 1 and -1. In matlab 'sin' library function is used to implement the sine function. The following graph represents the curve of sine function.

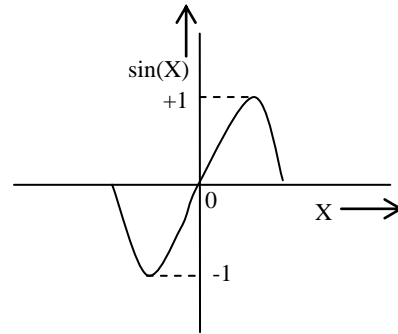


Fig 2: The graph of sine function

$$y = \sin(x)$$

ii) **Sigmoid Function:** This function is continuous function that varies gradually between the asymptotic values 0 and 1 or -1 and +1 as given by

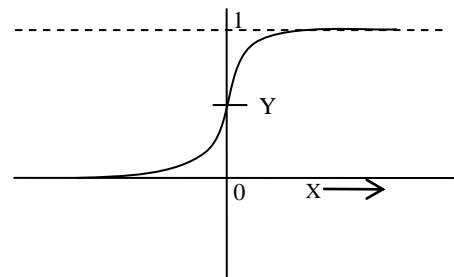


Fig 3: The graph of sigmoid function

$$y = \frac{1}{1 + e^{-x}}$$

iii) **Hard Limit Function:** When neurons uses hard limit transfer function the output they produces is 1 if net input reaches a threshold, otherwise it outputs 0. This allows a neuron to make a decision or classification. The Hard limit function can be implemented by library a function 'hardlim' in matlab.

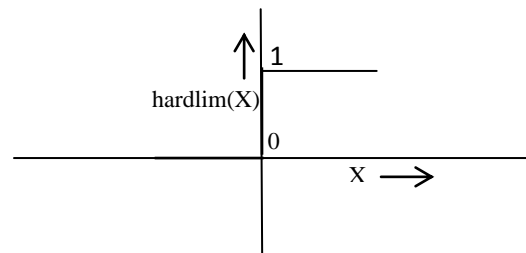


Fig 4: The graph of hard limit function

iv) **Radial Bias Function:** The outputs of radial bias function are based on the distance from some point called origin. The radial basis function can be implemented using library function 'radbas' in matlab. Following graph is representing the output curve for radial bias function.

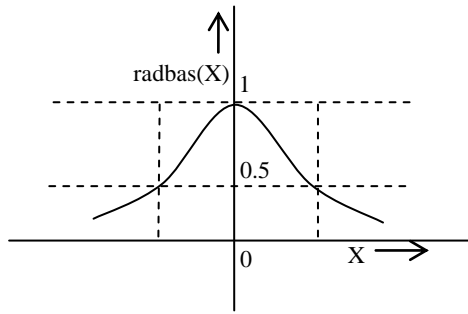


Fig 5: The graph of radial bias function

v) **Triangular Bias Function:** Triangular bias function can also be used as neural transfer function. This function calculates a layer's output from its given input. The triangular bias function can be implemented using 'tribas' library function in matlab. Following graph represent the curve of tribas function output.

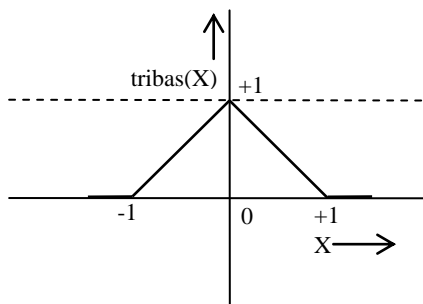


Fig 6: The graph of triangular bias function

4. EXPERIMENTAL RESULTS AND ANALYSIS

In this section the average performance of the activation functions are analyzed. The five activation functions, discussed in previous section, are used to obtain the result. Since the classifiers performances are measured by its accuracy of classification and time taken in testing by the classifier for an output [4] so the results are analyzed on the basis of accuracy percentage. The software environment that is used for simulation of ELM is MATLAB 7.8.0 (R2009a) on Intel core i3. The comparisons of experimental results are based on two major factors for calculating the accuracy of the ELM classifier. First is activation function and second is number of hidden neurons.

4.1 Dataset Description

For a classification of entity it is required dataset must be valid and complete. The selected datasets from UCI repository [13] are bupa, diabetes, ionosphere, iris, monk, spambase and yeast. These data sets are divided into two parts before classifying one for training and another for testing. Entities were selected randomly to create imbalance datasets for training and testing. The table given below shows the details of dataset in terms of number of records, training set, testing set and category (multiclass/binary).

Table 1. Dataset used for classification

Dataset	Category	Instances for Training	Instances for Testing	Total Instances
Bupa	Binary	183	162	345
Diabetes	Binary	561	192	753
Iris	Multiclass	79	71	150
Yeast	Multiclass	693	791	1484
Monk	Binary	124	124	248
Ionosphere	Binary	173	328	501
Spambase	Binary	2302	2299	4601

1.1 Experimental Results

For analyzing the performance of the ELM, the code was executed ten times with same number of neurons and activation function on single dataset. The execution for multiple times reflects randomization in the result. Due to randomization the average value is taken as output result. From table 2 to table 8 represent the average accuracy of for different dataset. Each table includes the average accuracy percentage in respect of number of hidden neurons and activation function. Both factors play a major role in classification of instances. The number neurons at hidden layer are taken for experiments are 10, 20, 30, 40 and 50.

Table 2. Average classification accuracy for bupa dataset.

Function	Number of Hidden Neurons				
	10	20	30	40	50
Sin	72.84	73.46	72.84	72.84	68.52
Sig	72.22	72.84	73.46	73.46	70.99
Hardlim	58.64	58.02	58.02	58.02	59.88
Tribas	69.14	70.37	69.75	69.75	69.14
Radbas	71.6	72.22	72.22	72.22	67.9

Table 3. Average classification accuracy for diabetes dataset.

Function	Number of Hidden Neurons				
	10	20	30	40	50
Sin	76.04	78.65	79.17	78.13	77.08
Sig	76.04	78.65	78.13	78.65	76.56
Hardlim	68.75	68.23	72.4	69.79	71.35
Tribas	77.6	75	78.13	79.17	77.6
Radbas	67.71	77.08	76.56	76.56	77.6

Table 4. Average classification accuracy for iris dataset.

Function	Number of Hidden Neurons				
	10	20	30	40	50
Sin	95.77	97.18	94.37	87.32	85.92

Sig	95.77	98.59	95.77	94.37	85.92
Hardlim	83.1	80.28	88.73	97.18	91.55
Tribas	87.32	98.59	98.59	98.59	98.59
Radbas	98.59	97.18	94.37	90.14	88.73

Table 5. Average classification accuracy for monk dataset.

Function	Number of Hidden Neurons				
	10	20	30	40	50
Sin	47.53	55.56	54.94	48.15	51.23
Sig	62.96	59.26	56.17	57.41	64.2
Hardlim	58.02	56.79	54.94	62.35	64.2
Tribas	43.21	47.53	45.68	48.15	53.7
Radbas	60.49	61.73	60.49	54.94	56.17

Table 6. Average classification accuracy for yeast dataset.

Function	Number of Hidden Neurons				
	10	20	30	40	50
Sin	57.02	58.28	60.56	59.54	59.04
Sig	57.4	59.67	58.53	59.17	59.17
Hardlim	43.11	48.42	49.81	52.09	52.47
Tribas	55.37	57.4	58.53	59.04	58.66
Radbas	57.14	58.79	60.05	60.05	60.3

Table 7. Average classification accuracy for ionosphere dataset.

Function	Number of Hidden Neurons				
	10	20	30	40	50
Sin	93.9	91.77	90.24	90.55	91.46
Sig	93.6	91.77	91.16	92.99	91.77
Hardlim	94.82	91.46	91.16	89.94	92.07
Tribas	92.99	91.16	89.63	91.77	89.33
Radbas	92.07	92.68	90.85	91.46	90.55

Table 8. Average classification accuracy for spambase dataset.

Function	Number of Hidden Neurons				
	10	20	30	40	50
Sin	69.38	74.95	76.69	78.51	80.56
Sig	70.25	73.64	76.56	78.03	80.73
Hardlim	61.11	62.03	65.77	65.03	64.72
Tribas	65.99	73.73	77.69	79.25	81.3
Radbas	69.86	73.38	76.99	79.82	80.21

The above results shows that the highest accuracy achieved by ELM is 98.59% in iris dataset it show that it is good classifiers but for datasets like monk its performance not more

that 60%. The performance graphs given bellow show the experimental summary for of ELM performance for different dataset

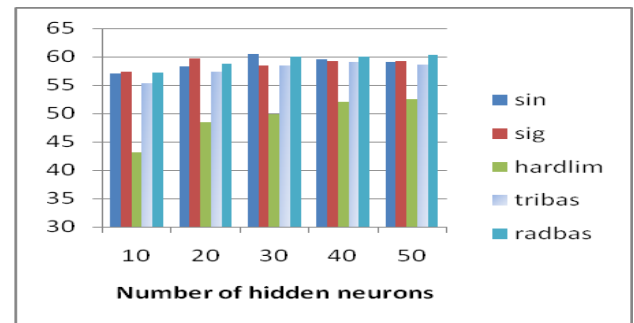


Fig 7: Summary of simulation between number of neurons and accuracy result for different activation function on yeast dataset.

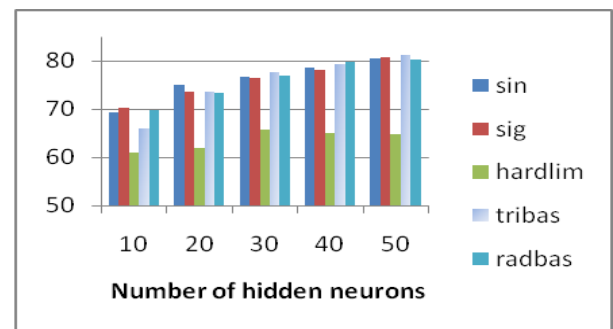


Fig 8: Summary of simulation between number of neurons and accuracy result for different activation function on spambase dataset.

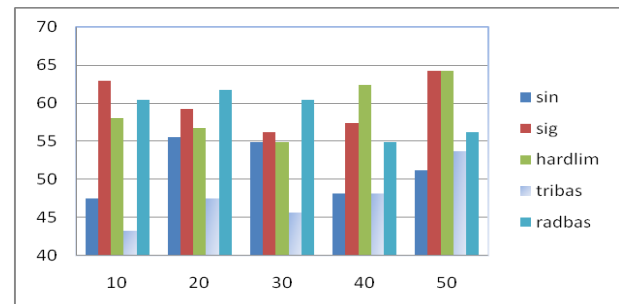


Fig 9: Summary of simulation between number of neurons and accuracy result for different activation function on monk dataset.

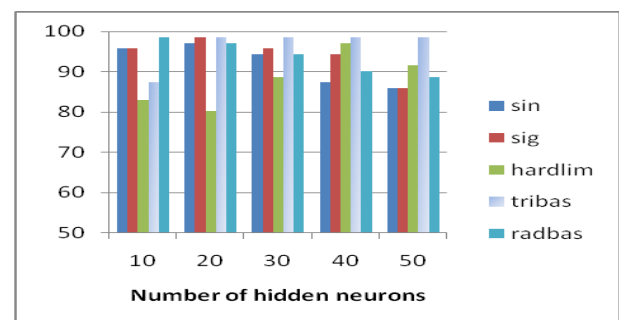


Fig 10: Summary of simulation between number of neurons and accuracy result for different activation function on iris dataset.

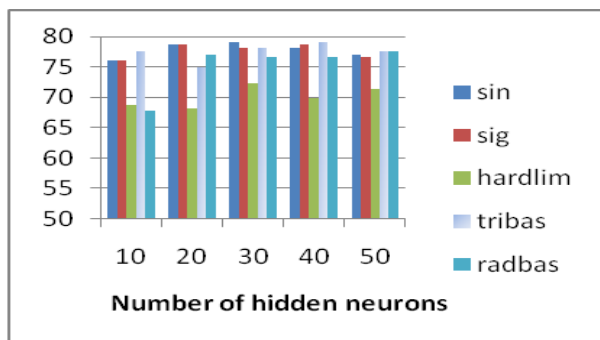


Fig 11: Summary of simulation between number of neurons and accuracy result for different activation function on ionosphere dataset.

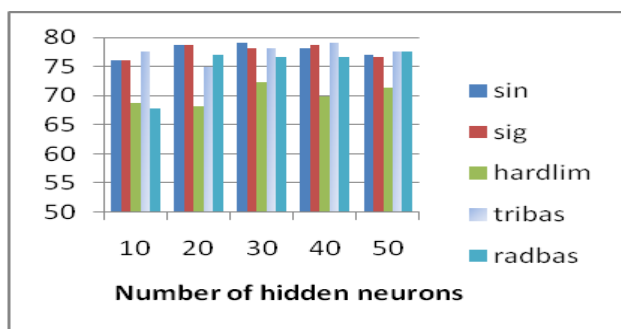


Fig 12: Summary of simulation between number of neurons and accuracy result for different activation function on diabetes dataset.

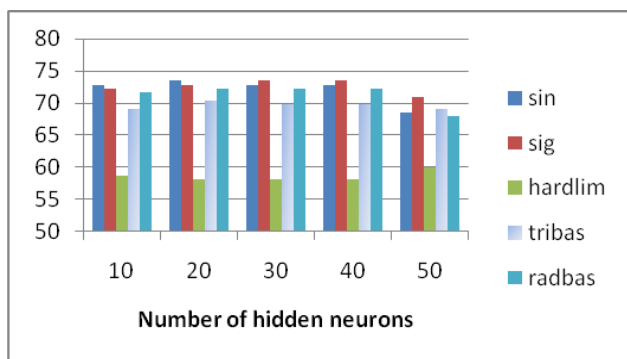


Fig 13: Summary of simulation between number of neurons and accuracy result for different activation function on bupa dataset.

The average testing accuracy is considered for the above results due to random accuracy percentage appeared on same dataset, same activation function and same number of hidden neurons. The following graphs represent the accuracy variations on different execution of ELM algorithm. The execution of code is performed for 10 times. The three dataset were selected to perform this accuracy variation test: Iris, Ionosphere and Bupa. The sigmoid (sig) activation function is used for the result on all three datasets.

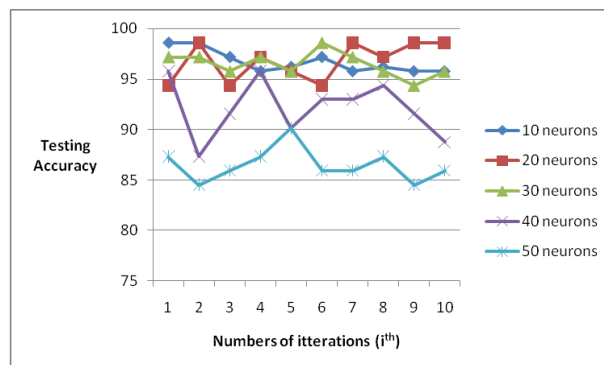


Fig 14: Graph between between testing accuracy and the 10 iterations of the algorithm execution on Iris dataset.

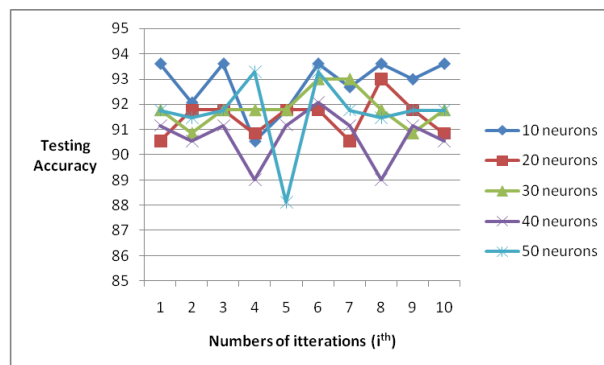


Fig 15: Graph between between testing accuracy and the 10 iterations of the algorithm execution on Ionosphere dataset.

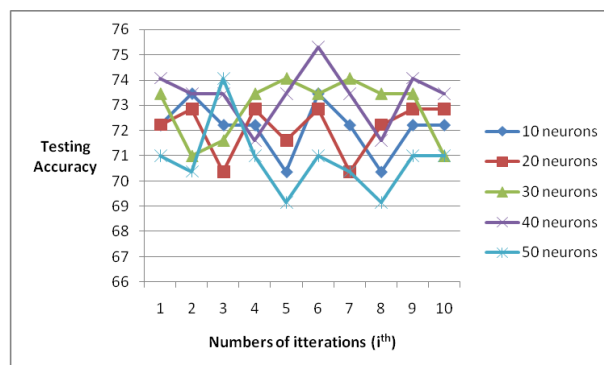


Fig 16: Graph between between testing accuracy and the 10 iterations of the algorithm execution on Bupa dataset.

In experimental result it is also observed that the reason of randomization for each iteration of ELM algorithm on same dataset with same number of hidden neurons and activation function due to calculations of input weights and bias values are performed through random number generating function. It gives different accuracy percentage for each iterations. The average training time take in training by the ELM is 0.0312 seconds for spambase dataset which contain largest instances for classification among all selected dataset therefore ELM is a fast classifier.

5. CONCLUSION

In this paper the comparison of activation function is performed with respect to different dataset and variation in neurons of hidden layer. It has analyzed that by changing the number of neurons of hidden layer the accuracy of classification is changing. It was observed that in dataset like

bupa, diabetes and yeast had a better performance on 40 neurons rather than 50 neurons. This yield that on increase in number of hidden neurons does not insures the accuracy of classifier. Another major factor of performance of classifier depends on activation function used for dataset result. This paper shows that not all activation function gives same result for any given dataset using same number of neurons. In a 'yeast' dataset sigmoid and radial bias functions were providing better accuracy for 30, 40 and 50 number of neurons. But in case of iris dataset triangular bias function is providing better accuracy than other function. The performance of hard limiter function in most of the dataset is less as compare to other activation function. This is due to property of activation function. The curve of hard limiter is not smooth so the values produced by each output neurons have very less difference which provides very low accuracy in classification. The other factors like input weight, size of input and bias value effects the performance of classifiers. Input weight and bias value are random numbers generated at each simulation so the accuracy varies accordingly. In future ELM classifier can perform much better if some approaches of data preprocessing may be added. These approaches can be a calculation of input weights by given dataset instead of using random number generating functions or some manipulation of data available in given dataset so that large intervals among data can be created for instances of different classes. This may reduce the randomization and increase the classification accuracy percentage.

6. REFERENCES

- [1] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, Rui Zhang, "Extreme Learning Machine for Regression and Multiclass Classification", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2011.
- [2] S.Tamura, M.Tateishi, "Capability and storage of four layered feedward neural network", IEEE Transaction on Neural Network, 1997.
- [3] Guang-Bin Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks", IEEE Transaction on Neural Network, 2003.
- [4] Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew, "Extreme learning machine: Theory and applications", Proceedings of International Joint Conference on Neural Networks, Budapest, Hungary, 2004.
- [5] Jim Y. F. Yam and Tommy W.S. Chow, "Feedforward networks training speed enhancement by optimal initialization of the synaptic coefficients", IEEE Trans. On Neural Networks, vol.12, no.2, pp. 430-434, March 2001.
- [6] Karayiannis and A.N. Venetsanopoulos, "Artificial neural networks: learning algorithms" performance evaluation, and application", Kluwer Academic, pp.135-158, 1993.
- [7] Y. LeCun, L.Bottou, G.B. Orr and K.R. Muller, "Efficient backprop" Lecture Notes in Computer Science, Vol.15, no. 24, pp.9-50,1998.
- [8] P. Lingras, C. Butz, "Rough set based 1-v-1 and 1-v-r approaches to support vector machine multi-classification", Information Science 177 (18) (2007) 3783-3798.
- [9] V. N. Vapnik, "The nature of Statistical Learning Theory," New York: Spriger-Verlag, 1995.
- [10] P.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973
- [11] S. Haykin, Neural Networks, A Comprehensive foundation, second ed, Pearson education Press, 2001.
- [12] S.k. Kay, Fundamentals of Statistical Signal Processing: Detection Theory, 1st ed., Prentice Hall, 1998.
- [13] G Wang and P Li, "Dynamic Adaboost Ensemble Extreme Learning Machine", 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE).
- [14] Guang-Bin Huang, Babri H A, "Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation function", IEEE Transaction on Nerual Network, 1998.
- [15] Guang-Bin Huang, Q.Y. Zuhu, C.K. Siew, "Extreme Learning Machine: A New Learning Scheme of Feedward Neural Network", 2004 International Joint Conference on Neural Networks, Budapest Hungar, 2004.
- [16] Guang-Bin Huang, Q.Y. Zuhu, C.K. Siew, "Extreme Learning Machine: Theory and Applications, Neurocomputing, 2006.
- [17] S.S.Haykin, "Neural Networks and Learning Machine", Prentice Hall, 2009
- [18] Khaled Fawagreh, Mohamed Medhat Gaber, Eyad Elyan, "Random forests: from early developments to recent advancements", Systems Science & Control Engineering: An Open Access Journal, 2014.
- [19] David Williams, Xuejun Liao, Ya Xue, Lawrence Carin, "Incomplete-Data Classification using Logistic Regression", 22nd International Conference on Machine Learning, Bonn, Germany, 2005.
- [20] Włodzisław Duch, Norbert Jankowski, "Survey of Neural Transfer Functions", NEURAL COMPUTING SURVEYS 2, 163-212, 1999.
- [21] Anguita, D., Ghio, A., Greco, N., Oneto, L., Ridella, S., "Model selection for support vector machines: Advantages and disadvantages of the Machine Learning Theory", The 2010 International Joint Conference on Neural Networks (IJCNN), 2010.
- [22] Erik Cambria, Guang-Bin Huang, "Extreme Learning Machines", IEEE Computer Society, 2013