ISSN 2320 - 2602

Volume 9 No.7, July 2020

International Journal of Advances in Computer Science and Technology

Available Online at http://www.warse.org/IJACST/static/pdf/file/ijacst19972020.pdf

https://doi.org/10.30534/ijacst/2020/19972020

A Survey on Various Extreme Learning Machines

Ambily Raveendran[1], Soumya Mathew[2]

[1]PG student, Department of CSE, Viswajyothi College of Engineering and Technology, Vazhakulam, India, ambilyremesh24@gmail.com

[2]Assistant Professor Department of CSE, Viswajyothi College of Engineering and Technology, Vazhakulam, India, soumya@vjcet.org

ABSTRACT

Feedforward neural networks (FNN) can be used for a broad range of machine learning applications. But the major disadvantage of FNN is its slower speed of functioning. This is for two reasons. Firstly, whole network parameters need to be adjusted frequently and secondly, the usage of gradient based optimization methods for learning, which is a slower algorithm. Apart from the conventional methods, this paper introducing a novel learning technique named Extreme learning machine (ELM). Here the parameters of hidden layers are generated arbitrarily and the weights of output layer are derived systematically, which leads to a fast training with low human intervention.

Key words: Extreme learning machine, ELM, Single-Hidden Layer Feed forward Neural Networks, SLFN.

1. INTRODUCTION

One of the neural networks that is most recommended is feedforward neural network (FNN).It can be constructed with a single input layer, single output layer many hidden layers. Input layer receives the stimuli from exterior domains and the network output can be transferred to the exterior domains through the output layer. A new set of rules is incorporated in a network called Single Hidden Layer Feedforward Neural networks (SLFN), which can be referred to as Extreme Learning Machine or ELM, where the dimensions of hidden layer are generated arbitrarily and the weights of output layer are derived systematically, which leads to a fast training with low human intervention. In traditional back-propagation training, weight adjustments and hidden layer adjustment can be performed by optimization processes based on gradient descend optimization. These training approaches are slow in general and at the same time it may reduce the generalization property of the network so that the result might be locked in local minima.

In extreme learning machine, randomly selected values are used for weights in the input layer as well as bias in hidden layer of the SLFN. The network output weights are calculated analytically by accepting the squared loss of the prediction error. ELMs tend to achieve the lowest possible learning error and the reduced output weight norm. For FNN, gaining a negligible learning error with minimal output weight rule leads to an outstanding generalization performance. Regardless of the matter, the persistence of the output of network hidden layer is established on arbitrarily allotted weights of input layer. Single hidden layer feedforward networks training based on ELM algorithm which have global approximators properties. Owing to its enormously fast learning speed, optimal generalization properties and the verified universal approximation/classification ability, ELM has efficiently gained finer learning accuracy in many applications like face recognition, text categorization, image classification etc. Nowadays different variations of ELM have been proposed, like neural-response-based ELM[1], online sequential ELM[2], error minimization based ELM[3], Optimally Pruned ELM[4], sparse Bayesian ELM[5], Semi-Supervised and Unsupervised ELM [6], sparse ELM[7], GEELM[8], HELM[9], Evolutionary cost-sensitive ELM[10] and MLK-ELM[11].

The remaining sections of this paper are arranged according to the following. The basics of Extreme Learning Machine is discussed in section 2 .Section 3 deals with the study on various extreme learning machines. Section 4 and Section 5 gives some discussion and conclusion of this paper.

2. FUNDAMENTALS OF EXTREME LEARNING MACHINE

In this segment, we describe about the basics of ELM. Initially, Huang et al. [12], [13] introduced ELM in SLFNs, later applied in generalized SLFNs.

The key points of ELM are given below:

i. Repetitive tuning in hidden layers of ELM is not essential [12], [15].

ii. As per FNN theory, the norm of weights $\|\beta\|$ and error occurred during training $\|H\beta - T\|$ are need to be minimized [12], [15].

iii. Before performing hidden layer feature mapping, universal approximation condition [13], [14] need to be satisfy.

In ELM, the main idea focuses on the weight adjustment of hidden layer. In addition, the biases are arbitrarily produced and the calculation of the weights performed in output layer. ELM's basic architecture is as shown in figure.1.

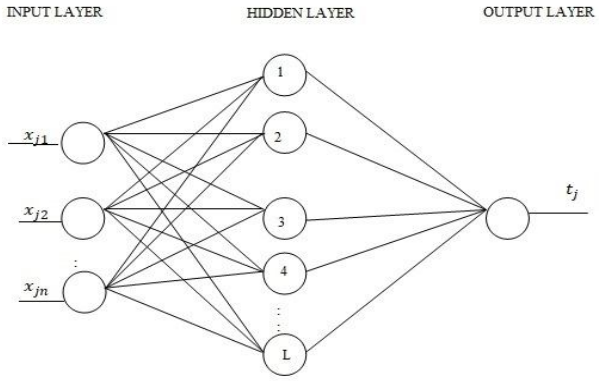

Figure 1: ELM Architecture

The SLFNs output function with L hidden neurons can be expressed as,

$$f_L(x_j) = \sum_{i=1}^{L} \beta_i g_i(x_j) = \sum_{i=1}^{L} \beta_i G(a_i, b_i, x_j) = t_j \ , \ x \in R^d , \beta i \in R^m \quad ,j=1,2,..N \qquad (1)$$

Where g_i represents the output function $G(a_i, b_i, x)$ of the i^{th} hidden neurons.

The activation function g, for additive neurons g_i can be expressed as ,

$$g_i = G(a_i, b_i, x) = g(a_i . x + b_i) \ , \ a_i \in R^d , b_i \in R \qquad (2)$$

The activation function g, for the nodes with radial basis function (RBF) g_i can be expressed as,

$$g_i = G(a_i, b_i, x) = g\left(b_i \| x - a_i \|\right), \quad a_i \in R^d , b_i \in R^+ \qquad (3)$$

Equation (1) can be remodify as

$$H\beta = T \qquad (4)$$

Where $H(a_1,...a_L, b_1,..b_L, x_1,...x_N) =$

$$\begin{bmatrix} G(a_1,b_1,x_1) & .. & G(a_L,b_L,x_1) \\ \vdots & :: & \vdots \\ G(a_1,b_1,x_N) & .. & G(a_L,b_L,x_N) \end{bmatrix}_{NXL}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{LXm} \quad \text{and} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{NXm}$$

β^T is the transpose of vector β . The entire Matrix **H** represents the output of hidden layers , the i^{th} column of matrix **H** represents the i^{th} hidden neuron's output vector in accordance with the inputs and the j^{th} row of **H** represents the output vector of the hidden layer .

In order to compute the output weights β we require the knowledge of the hidden layer output matrix **H** and the target values. Here the target values are obtained from Moore-Penrose generalized inverse of the matrix as H^+ . ELM algorithm can be summarized as follows:

Algorithm:

Consider the training set as (x_i, y_i), $x_i \in R^{d1}$, $y_i \in R^{d2}$, the activation function as $f : R \to R$, and the number of hidden neurons as N:

Step1. Weights of input layer w_i and biases b_i are randomly assigned, where $1 \leq i \leq N$.

Step2. Calculate **H**, where **H** represents output matrix of the hidden layer.

Step3. Calculate $\beta = H^+ T$ **,** which represents the matrix of output weights.

3. LITERATURE SURVEY

Li et al. [1] suggested a new technique for extraction of features named extreme learning machine based on neural response (NR-ELM), influenced by neural-response model. The NR-ELM model is a hierarchical method with two phases, a multi layer ELM mapping phase and an ELM training phase. In the first phase two operations are performed recursively and alternatively. They are feature map generation and max pooling operation. To improve the invariance properties of the entire system, initially the input

image is passed through a dense SIFT preprocessor, which will extract strong features of that input image and the output will be the new input to ML-feature mapping phase. During each feature map construction layer, input weights are randomly selected and used with activation functions that fulfilling the theory of ELM. To extract the most different latent information during pooling operation, the maximum operation is executed. In this brief, two layer of max pooling operations are performed. In first layer , finding the local maxima and in second layer detecting the global maxima. During the ELM learning phase output weights are learned using elastic-net regularization algorithm which assists to train more meaningful and compact output weight. As ML-ELM is built by layer wise stacking of ELM-based auto encoders, NR-ELM is a bit slower than the ML-ELM. Also to the last layer input no random feature mapping is performed. Sparse-based NR algorithm has an outstanding rotation invariance property by proposing sparse coding to the NR framework and it is consistently perform better than NR-ELM when images are rotated with different maximal angles.

Liang et al. [2] introduced a new online sequential ELM algorithm (OS-ELM) for single hidden layer feedforward neural networks based on ELM and recursive least square algorithm (RLS). OS-ELM is derived from batch learning training methods. This can consider additive nodes as well as RBF within a single system. Here the OS-ELM will learn the training data either individually or as chunk-by-chunk. Chunks may be in predetermined size or in varying size. In OS-ELM, it discards the samples for which the learning has been previously performed. This algorithm has no underlying information with respect to the number of training observations will be exhibited. For sequential data learning, OS-ELM contains two stages. In the initialization stage, learning with a chunk from an existing training data and in the second phase, which is a sequential learning phase, in which learning is performed with newly collected samples. In first stage, online sequential ELM uses the conventional extreme learning machine to learn the SLFN whereas in second stage, the matrix for the network's output weight is modified with Recursive Least Square algorithm by utilizing the newly obtained data. The weights between the input neurons and the hidden neurons as well as biases are arbitrarily generated. The output weights can be calculated systematically with additive nodes. In the same way, the architecture with RBF nodes, widths of the neurons and the centers are randomly generated as well as fixed and output weights are analytically calculated. Usually tuning of various control parameters can be done in sequential learning algorithms. But in OS-ELM we required to identify only the hidden nodes number. While comparing with other popular sequential learning techniques, the proposed OS-ELM generates more desirable generalization performance in a quick learning rate. Main highlight of this approach is, its adaptabilities in processing various sequential data volume, i.e., OS-ELM can deal with newly obtained data as individually and chunk-by-chunk with predetermined or varying length.

An extreme learning machine based on error minimization technique that automatically define the hidden neuron numbers is recommended by Feng et al. [3]. Here hidden layer are developed by adding either single nodes as one by one or group of nodes as batch by batch. This will repeat until the corresponding output error reaches the minimum. After adding each set of hidden nodes, EM-ELM revise the output weights incrementally each time which will reduce the computational complexity.

When we deal with correlated or irrelevant data, original ELM cannot be considered as an efficient algorithm. So Miche et al. [4] introduced a new methodology known as, Optimally-Pruned-ELM (OP-ELM). In this approach perform pruning of neurons in the network for getting more generic and robust algorithm. During the primary step, construct an MLP using conventional ELM with a large network. Build an MLP with a large number of nodes in the primary phase using traditional ELM. Then the nodes are ranked based on their relevance by utilizing Multi response sparse regression algorithm (MRSR). In the last step, using Leave One Out (LOO) approval method selects optimal neuron numbers and prunes the remaining. In this model, a blend of three unique kinds of kernels linear, Gaussian and sigmoid kernels are used for more generality. OP-ELM is more effective than conventional ELM in dealing with correlated or irrelevant data.

Luo et al. in brief [5] suggested a novel approach, SBELM, to find the output weights of ELM classifier. Instead of using Moor-Penrose generalized inverse, here exploiting Bayesian method to learn output weight. In this model, in lieu of single shared prior for entire weights, individual regularization priors are used on one and all weight. So some weights are automatically adjusted to zero, especially weights with high regularization priors. Those hidden-nodes analogous to these null weights can then be trimmed, directing to a sparse network. Pruning of hidden neurons in learning phase improves the robustness. With the SBELM dominant properties, more dense, rapid and precise engine framework can be constructed. SBELM takes over the velocity of learning from the ELM method and the weight's sparsity from the sparse Bayesian learning technique. Best part of this approach is, it is comparatively insensitive to the initial hidden-neuron numbers.

Huang et al. suggested a unified model based on manifold regularization for both semi-supervised and unsupervised ELM in [6]. Here the unsupervised (unlabeled) or

semi-supervised (partially labeled) samples are clustered using ELM. For both algorithms, semi-supervised and unsupervised problems, share same set of stages for hidden layer generation and this is the crux of the ELM theory. The distinction lies in the next phase on how the output weight calculation performed. The recommended semi-supervised extreme learning machine (SS-ELM) utilizes unlabeled data to enhance accuracy of classification. SS-ELM focuses to resolve the output weights by diminishing the generalized least square estimate regularization cost function. To get the embedded matrix for clustering, the proposed unsupervised ELM (US-ELM) employs least square method. Here, k-means clustering algorithm can be used to execute clustering in embedded space. Both algorithms fabricated their work on two beliefs: the first one is, all data have identical borderlines distribution i.e., the manifold regularization belief and the second one is smoothness assumption. In a nutshell, if we consider the samples as a graph in the training set, there should not be countless variations between neighboring instances, rather it should be smooth.

Both kernel and hidden nodes worked together in [7] sparse ELM, which is a blend of various classification techniques like SLFNs, support vector machine and radial basis function networks. As sparse ELM supplies more compressed network as compared to unified-ELM, which diminishes time complexities and storage space. The learning of sparse ELM is basically a Quadratic Programming Problem like SVM training. A complex QP problem is break up into sequence of small sub-problems. The main contradiction between them is, SELM doesn't have the sum constraint and only one Lagrange variable is modified during each time. Thus the training procedure would be much simple. In sparse ELM, iterative computations are the basis of the learning algorithm, while in unified-ELM solution it is determined based on inverse matrix. Accordingly sparse ELM learning speed is better than unified-ELM. Sparse-ELM is indeed a better solution to problems of growing scale like data compression, image processing, neuroscience etc. Unified-ELM resolves multi-class problems directly. But here need to merge a number of binary sparse ELM together. So this approach is less beneficial than unified extreme learning machines in applications with multiclass. Another drawback is, accuracy deviation in testing and learning phases for multiclass problem is too higher than that of unified ELM.

In paper [8] Iosifidis et al. recommended a new version of ELM algorithm, GE-ELM, which is capable of using penalty as well as intrinsic subspace learning (SL) criteria to optimize the output-layer weights of ELM. Here exploiting formulations based on supervised as well as unsupervised SL criteria in optimization of ELM. In graph embedding framework, exploiting the SL techniques such as Laplacian Eigenmaps (LE), Linear- Discriminant Analysis (LDA), Marginal Disriminant Analysis (MDA), Locally Linear Embedding (LLE), Marginal Fisher Analysis and Local Fisher Disriminant Analysis (LFDA). Here generating a graph Laplacian matrix to acquire the local geometric structure of the input sample and enhance the generalization capability of the ELM algorithm. The major disadvantage of this approach is GEELM approach exploiting only the local geometric information of input sample. The local and global geometric information of the input sample can effectively improve the identification effect of the ELM algorithm.GEELM method using local geometric information only. So GLELM is better than GEELM since it considers local as well as global geometry.

A novel hierarchical-ELM based deeper network structure was introduced in [9] by Tang et al. in which multiple auto-encoders are used as hidden layers. Unlike conventional DL techniques, the hidden layer output in hierarchical-ELM does not require to train the model in the greedy layer-wise manner, i.e., here extraction of features and classification are two independent portions of the system. So the best solution can be gained by one-shot learning. Consequently hierarchical-ELM required lesser training time than other DL algorithms. The learning system of hierarchical-ELM includes two independent phases: In first phase, learn features using unlabeled data, here a new ELM-based auto-encoder are tries to fine tune the inputs to make the reconstructed output data being equivalent to the input data. In second phase, i.e., supervised classification phase, the classical ELM based regression is executed for ultimate classification process. In this Hierarchical-ELM approach, the auto-encoder is a self-encoder with sparse constraints. In this approach initially the raw input data converted into a randomly selected feature-space of ELM. This can enable to utilize hidden-features among sample dataset. Afterwards, an unsupervised training with N-layers should be implemented to get prominent sparse features. Different from automatic-encoder in conventional DL approaches, the weights of input of ELM sparse auto-encoder is entrenched by finding a path back from a randomly selected space. Theory of ELM exhibits that ELM learning with randomly selected input weights can estimate any input samples. That is to say, if the auto-encoder is learned based on ELM theory, once the weights of auto-encoder is initialized, no need to fine-tune the parameters and it will be fixed. So as compared to conventional deep learning methods, it has better training proficiency. An ELM-based sparse auto-encoder generated through l_1 norm optimization is utilized to produce multiple layer sparse-features of the input sample, that attempts to generate the reconstructed sample data similar to it and acquire much better hidden features. Hierarchical-ELM can further prune the neurons count in the hidden-layer and thus stimulate the training process. Experiments on different datasets exhibit that Hierarchical-ELM method attains

improved and quicker classification than the available hierarchical learning algorithms. In H-ELM classification is performed on sparse representation of input samples, but in original ELM, classification executed on raw input data directly. Hence hierarchical-ELM reaches better performance than classical ELM. But main disadvantage is that, efficiency of Hierarchical-ELM is too sensitive to the regularized LMS computation parameters than that of original ELM.

Zhang et al. [10] introduced ECSELM method to enhance performance of ELM for cost sensitive tasks. Currently existing recognition systems based on ELM accomplish small error rate by guessing an equal loss for any misclassification. The dissimilar losses in a Facial Recognition method have been primarily focused by constructing a cost sensitive classification problem. In the specified Evolutionary Cost Sensitive-ELM, optimized cost-sensitive training is organized for managing the similar problem of loss in extreme learning machine. The loss can reduce by cost-sensitive training using a pre-calculated cost matrix which evaluates the criticality of one category of misclassification over another. Here we have to find the optimal cost matrix, where a patterned cost-sensitive behavior exists to achieve a better classification efficiency. On the basis of Cost Sensitive-ELM, the Evolutionary Cost Sensitive-ELM is to search the optimized cost-matrix, that results an excellent classification through the output layer weights with reference to cost matrix, so that acquires a minimum value for the total loss between the actual value and the predicted value. The cost matrix is normally resolved in a practical way and for sensitive learning, that may cause low generalization performance. We can resolve this by optimizing the cost-matrix using an evolutionary algorithm called Backtracking Search optimization Algorithm. To create trial individuals in this search technique, three fundamental systems like selection operator, mutation operator and crossover operator can be used. This easy and fast method is useful to solve multimodal problems.

Wong et al. presented a kernel variety of Multilayer-ELM, which is multilayer kernel ELM in [11]. ML-KELM endorses two separate learning techniques, K-ELM and ML-ELM. Two phases of ML-KELM are a) Unsupervised representation learning by stacked KELM-AE (kernel variety of ELM-AEs) b) Supervised feature classification using a K-ELM (kernel type of ELM). The multilayer kernel ELM does not required parameter tuning for any layer like multilayer-ELM. Also to obtain optimal model generalization, there is no random projection techniques are used. Instead, multilayer kernel-ELM learns an optimized pattern from one or only few samples under constant parameters. For getting a smaller reconstruction error, instead of using pseudo inverse, learned using exact inverse of kernel matrix and pay off rebuilding of data representation with little error and reaches finer model generalization. Since all transformation matrices are integrated into two matrices, store space can be diminished and may shrink execution time of model. Provide better model generalization since it is using exact inverse of kernel matrix instead of pseudo inverse. As it is using transformation matrix for entire layers clubbed into two matrices, storage size are fixed.

4. DISCUSSION

This paper focuses on various extreme learning machine methods. Each of these methods has its own advantages and disadvantages. Main merits and demerits of each method are shown in Table 1.

Table 1: Advantages and disadvantages of different ELM approaches

Reference	Approach	Advantages	Disadvantages
[1]	Extraction of features using NR-ELM for classification	• Attains the best classification accuracy. • Suitable for practical big data problems. • Good computational time over conventional DL algorithms.	• It is little slower than Multilayer-ELM. • Rotation invariance property of NR-ELM is not good as SNR algorithm when inputs are rotated with various maximal angles.
[2]	OS-ELM for SLFN based on ELM and RLS.	• Low computational complexity. • Flexible to handle various size data.	• Not able to regulate input weights. • It is not suitable to train Recurrent Neural Network.
[3]	Automatically define the hidden neuron numbers using EM-ELM	• While comparing error minimization ELM with I-ELM and EI-ELM, obtain equal or higher generalization	• EM-ELM producing more fluctuating results compared to original ELM.

		performance and faster training speed.	
[4]	OP-ELM perform pruning of neurons in the network for getting more generic and robust algorithm	• More efficient in handling irrelevant or correlated data. • Good accuracy with small computational time.	• It is slower than original elm.
[5]	SB-ELM exploiting Bayesian method to learn output weight of ELM classifier	• Insensitive to the initial hidden neuron number. • Best performance at small hidden neuron number. • Produce more compact model. • High generalization, Low computational cost and fast training time.	• In the SB-ELM, the parameters of the Bayesian inference method as well as ELM are correlated, and must be trained together through an iterative optimization method. • SBELM cannot handle with missing data.
[6]	Unified model based on manifold regularization for both semi-supervised and unsupervised ELM	• Can handle multi-class classification or multi-cluster clustering. • Can manipulate unseen data directly during test time. • Require less learning time on multi-class classification problems.	• When the level of noise increased to 50%, performance will be poor than that on clean data in US-ELM.
[7]	Sparse-ELM for classification is a blend of	• Favourable for grow-scale	• Less beneficial as compared to
	various techniques like SLFNs, support vector machine and radial basis function	tasks owing to faster speed of learning as well as testing. • Minimum space for storage is required. • For large problems training speed will be faster than unified ELM.	unified ELM in multi-class problems. • High variation in learning and testing accuracy for multi-class.
[8]	GE-ELM exploiting the penalty as well as intrinsic SL criteria to optimize the output-layer weights of ELM	• Better performance than conventional ELM (inclusion of subspace learning criteria revealing intrinsic as well as penalty data relationship improves performance).	• GEELM method use only local geometric information for recognition. GLELM is better than GEELM since it considers local as well as global geometry.
[9]	In H-ELM, multiple auto-encoders are used to fine tune inputs and classical ELM based regression for ultimate classification process	• Speed as well as accuracy of training is high. • Better performance than original ELM.	• The performance of hierarchical-ELM is more parameter sensitive.
[10]	ECSELM method to enhance performance of ELM for cost sensitive tasks	• It well explains cost-matrix definition issue in cost-sensitive learning problems. • Classification of a given instance is possible without	• Not applicable for class imbalance learning (CIL) problems.

		multi-class voting technique stated in SVM.	
[11]	ML-KELM endorses two separate learning techniques, K-ELM and ML-ELM	• No need parameter tuning for all layers. • Provide better model generalization since it is using exact inverse of kernel matrix. • Fixed storage size.	• Need high storage space and very slow learning time.

5. CONCLUSION

The main focus of the paper is to study different ELM variants. Each method has its own advantages and disadvantages. As a learning method, ELM has established excellent capabilities to solve classification and regression problems. Nowadays ELM methods have received much surveillance in machine learning as well as computational intelligence groups, in both theoretical study and implementations. Basics of extreme learning machine methods are comprised of universal approximation capability with arbitrary selected hidden-layer as well as different training algorithms with easy and fast implementations.

REFERENCES

1. H. Li, H.Zhao and H. Li, **Neural-Response-Based Extreme Learning Machine for Image Classification**, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2. pp. 539–552, 2019.
2. N. Y. Liang, G. Bin Huang, P. Saratchandran, and N. Sundararajan, **A fast and accurate online sequential learning algorithm for feedforward networks**, *IEEE Transactions on Neural Networks*, vol. 17, no. 6. pp. 1411–1423, 2006.
3. G. Feng, G. Bin Huang, Q. Lin, and R. Gay, **Error minimized extreme learning machine with growth of hidden nodes and incremental learning**, *IEEE Transactions on Neural Networks*, vol. 20, no. 8. pp. 1352–1357, 2009.
4. Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutte and A. Lendasse, **OP-ELM: Optimally pruned extreme learning machine**, *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 158–162, Jan. 2010.
5. J. Luo, C. M. Vong, and P. K. Wong, **Sparse bayesian extreme learning machine for multi-classification**, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 4. pp. 836–843, 2014.
6. G. Huang, S. Song, J.N.D Guptha and C.Wu **Semi-supervised and unsupervised extreme learning machines**, *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2405–2417, 2014.
7. Z. Bai, G.-B. Huang, D. Wang, H. Wang and M. B. Westover, **Sparse Extreme Learning Machine For Classification**, *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp.1858-1870, 2014.
8. A. Iosifidis, A. Tefas, and I. Pitas, **Graph embedded extreme learning machine**, *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 311–324, 2016.
9. J. Tang, C. Deng, G. B. Huang, **Extreme learning machine for multilayer perceptron**, *IEEE Transactions on Neural Networks and Learning Systems* , vol. 27, no. 4, april 2016.
10. L. Zhang, D. Zhang, **Evolutionary Cost-Sensitive Extreme Learning Machine**, *IEEE Transactions on Neural Networks and Learning Systems* PP (99) (2016), 1–16.
11. C. M. Wong, C. M. Vong, P. K. Wong, and J. Cao, **Kernel-Based Multilayer Extreme Learning Machines for Representation Learning**, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 3. pp. 757–762, 2018.
12. G-B. Huang, Q-Y. Zhu, C-K. Siew, **Extreme learning machine: a new learning scheme of feedforward neural networks**. *In: Proceedings of international joint conference on neural networks (IJCNN2004)*, vol 2, Budapest, Hungary, 25–29 July 2004, pp 985–990
13. G-B. Huang, L. Chen, C-K. Siew, **Universal approximation using incremental constructive feedforward networks with random hidden nodes,** *IEEE Transactions on Neural Networks* 17(4):879–892
14. G-B. Huang, L. Chen, **Enhanced random search based incremental extreme learning machine**, *Neurocomputing* 71:3460–3468
15. G-B. Huang, Q-Y. Zhu, C-K. Siew. **Extreme learning machine: theory and applications,** *Neurocomputing* 70:489–501