Dynamic Adaptation for Length Changeable Weighted Extreme Learning Machine

Conference Paper · December 2019

1 author:

Tarek Berghout
Université Batna 2

This work is submitted to international conference of intelligent systems in 09.12.2019. The toolbox of The proposed PW-LCI-ELM is publicly available in this link:
https://www.mathworks.com/matlabcentral/fileexchange/73629-rul-prediction-c-mapss dataset?s_tid=prof_contriblnk
Please cite this work as:
BERGHOUT Tarek, Mouss Leila Hayet, Kadri Ouahab, "Dynamic Adaptation for Length Changeable Weighted Extreme Learning Machine", International Conference of Intelligent systems (2020), Amsterdam, The Netherlands, submitted: 9.12. 2019, accepted 23.12.2019.

Dynamic Adaptation for Length Changeable Weighted Extreme Learning Machine

Berghout Tarek[1] , Mouss Leila Hayet[1], Kadri Ouahab[2].
[1]Laboratory of Automation and Manufacturing Engineering
University of Batna2, 01 Rue Chahid Bokhlof, Batna, 05000, Algeria,
berghouttarek@gmail.com, hayet_mouss@yahoo.fr
[2] Department of Computer Science, University of Batna 2
53, Fesdis, Batna 05078, Algeria
o.kadri@univ-batna2.dz

Abstract. In this paper, a new length changeable extreme learning machine is proposed. The aim of the proposed method is to improve the learning performances of the Single hidden layer feedforward neural network (SLFN) under rich dynamic imbalanced data. Particle Swarm Optimization (PSO) is involved for hyper-parameters tuning and updating during incremental learning. The algorithm is evaluated using a subset from C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset of gas turbine engine and compared to its derivatives. The results prove that the new algorithm has a better learning attitude. The toolbox that contains the developed algorithms of this comparative study is publicly available.

Keywords: Extreme Learning Machine, C-MAPSS dataset, Particle Swarm Optimization, turbofan engine.

1 Introduction

Length Changeable Extreme Learning machine LCI-ELM is one of Extreme Learning Machine (I-ELM) variants proposed in [1] in the aim of founding the suitable architecture of the SLFN under any application. Unlike original Incremental ELM (I-ELM) [2] that allows the hidden nodes to grows one-by-one, LCI-ELM is proposed to accelerate the addition of hidden nodes group-by-group according to user specifications. The main problems of LCI-ELM in its original version are: (1) the over-fitting problem is not discussed. (2) Adaptation of the algorithm under "time varying" data is not aborted.

The common solutions of over-fitting problems in ELM are generally depends on Tikhonov (L2 norm minimization) or LASSO (L1 norm minimization) regularization or both as mentioned in [3]–[6]. The choice of the optimal regularization parameters in most of ELM variants is usually based on Leave One Out cross validation using Allen's prediction sum of squares known as (MSE^{PRESS}). The MSE^{PRESS} is determined under sparse representations which experimentally outperforms the LOO

This paper submitted in: 09.12.2019, Accepted in: 23.12.2019.

under ordinary feature mapping to hidden layers. However LOO methods allows only to peak up the best regularization parameter from a set of chosen ones without any improvements or dynamic adaptation.
Concerning the second problem, the imbalanced data adaptation is usually done by adding weighting parameters to the pseudo inverse method to guarantee an optimal trucking of data variations.

Through this brief analysis, our proposed solutions for the above mentioned problems are: (1): allowing the regularization parameters to adapt dynamically to the incremental learning. (2) Using weighted ELM theories to satisfy dynamic changes of training data[7]. These two solutions are carried out using PSO algorithm which has the ability to enhance and to select the optimal hyper-parameters in the same time.

The proposed algorithm PW-LCI-ELM is compared to its LOO derivative LR-LCI-ELM and original LCI-ELM using dynamic data retrieved from C-MAPSS software[8]. The details of the dataset are not aborted in this paper, the one can found them in [9]. The results proves that the new enhancements achieved a better adaptation of the training weights towards “time –varying” data. MATLAB codes of this comparative study are well organized in single toolbox which is publicly available in[10].
This paper is organized as follow: The description of the proposed algorithm is presented in section 2. The results with discussion are showcased in section 3. This paper is concluded in section 4.

2 Methods

ELM firstly was introduced in[11] for training an SLFN, and then extends to fit any type of learning with a verity of architectures. For a given training set $\{X,T\}$ where (X) and (T) represents its inputs and targets respectively. The training rules of ELM for the minimization problem in “(1)”are:

$$E = \left\| H\beta - T \right\|^2 \tag{1}$$

- Randomly generate the hidden nodes parameters, weights (a) and biases (b).
- Calculate the hidden layer (H) using any activation function (G) depending on “(2)”.

$$H = G(aX + repmat(b)) \tag{2}$$

- Determine analytically the output weights of the SLFN using “(3)”.

$$\beta = pinv(H)T \tag{3}$$

The number of hidden neurons is specified from the user and tuned manually according to its experience.

I-ELM is introduced in [2] attempting to fined the suitable architecture of the SLFN by adding hidden nodes one-by-one. However this iterative searching consumes more computational costs. Therefore, LCI-ELM appears with a new incremental construction paradigm by adding hidden nodes group-by-group to accelerate the search mechanism based on “(4)”.

$$l_i = ke^{-\lambda(i-1)} + 1 \tag{4}$$

Where (λ) and (k) are user specified parameters controls the speed of convergence of growth in hidden nodes (l_i) towards the maximum number of hidden nods $(l_{\max})$ for each iteration (i).

On the one hand, our contribution in this works is by adding a weighting parameter (W) and regularization parameter (C) to reduce over-fitting and to adapt the network towards to different data changes using “(5)” instead of “(3)”[7].

$$\beta = \begin{cases} pinv(H^T WH + CI)H^T WT, N \geq l. \\ H^T pinv(WHH^T + CI)WT, N \leq l. \end{cases} \tag{5}$$

N Represents the number of training data and l is number of used hidden nodes.

On the other hand, we used PSO algorithm [12] to search and update the two parameters in every incremental learning iteration of LCI-ELM.

By introducing the regularization parameter to LCI-ELM algorithm. The objective function of PSO algorithm defined as shown in “(6)”.

$$E = \left\| H\beta - T \right\|^2 + C\left\| \beta \right\|^2 \tag{6}$$

3 Experimental results and discussion

The comparative study is preceded under higher dimensional “time-varying” data. The training and testing sets are retired from the FD001 subset of the C-MAPSS dataset.

Two essential metrics are generally used to evaluate “data-driven” approaches in this benchmark. The RMSE in “(7)” and the score function “(8)”[13].

$$s = \begin{cases} \sum_{i=1}^{N} e^{-d/13} - 1, \text{ for } d_i < 0 \\ \sum_{i=1}^{N} e^{-d/10} - 1, \text{ for } d_i \geq 0 \end{cases} \tag{7}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} d^2} \tag{8}$$

d is the difference between the estimated and desired target.

The results of this study are illustrated in Table1.

Table 1. Experimental results evaluation

Methods	Training		
	Time	RMSE	Hidden nodes
LCI-ELM	3.7284	15.3693	35
LR-LCI-ELM	0.8736	15.9509	23
PW-LCI-ELM	12.5893	15.9220	26
Methods	**Testing**		
	Time	RMSE	Score
LCI-ELM	0.0312	49.4898	1.6249e+06
LR-LCI-ELM	0.0012	48.0103	1.1755e+06
PW-LCI-ELM	0.0468	45.9682	9.8030e+05

The results from Table 1, confirm that the proposed approach can achieve better generalization with an acceptable number of neurons using adaptive balancing and dynamic regularization.

The variations of (W) and (C) during random search based PSO are illustrated in Fig.1.

The stooping criteria in Fig.2. Indicates that the dynamic adaptation of (W) and (C) allows the networks to extend more towards better generalization.

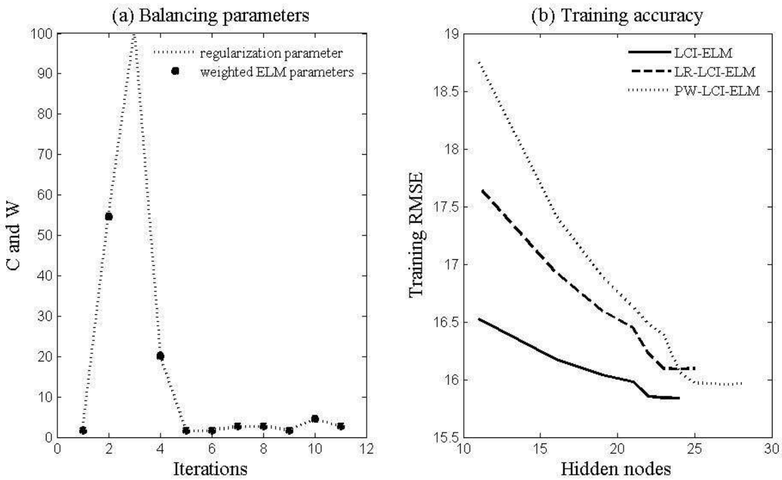


Fig. 1 Algorithms evaluation under incremental learning iterations

The scoring functions in Fig.2 behave with less sparseness towards higher error values. This explains the effect of adaptation during training.

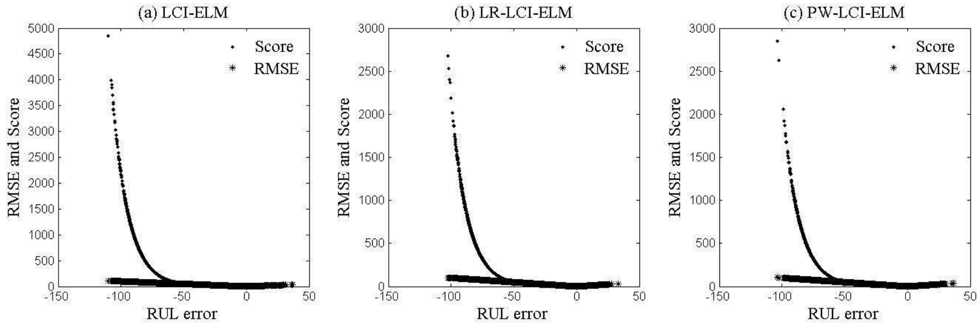


Fig.2. Scoring function comparison

4 Conclusion

This work introduces a new dynamic balancing and regularization paradigms for the incremental construction of an SLFN under "time-varying" data.

The proposed algorithm is studied and compared to its derivatives from the literature and proved its accuracy.

Only one type of regularization (Tikhonov) under full rank feature mapping is discussed. Therefore the aim of future works will focus on more regularization norms and new feature representation such as kernels or sparse mappings using SLFNs.

References

1 Y. X. Wu, D. Liu, and H. Jiang, "Length-Changeable Incremental Extreme Learning Machine," *J. Comput. Sci. Technol.*, vol. 32, no. 3, pp. 630–643, 2017.

2 G. Bin Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16–18, pp. 3056–3062, 2007.

3 Y. Miche, M. van Heeswijk, P. Bas, O. Simula, and A. Lendasse, "TROP-ELM: A double-regularized ELM using LARS and Tikhonov regularization," *Neurocomputing*, vol. 74, no. 16, pp. 2413–2421, 2011.

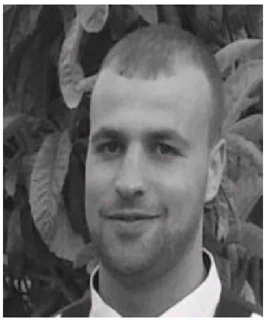
4 M. W. Fakhr, E. N. S. Youssef, and M. S. El-Mahallawy, "L1-regularized least squares sparse extreme learning machine for classification," *2015 Int. Conf. Inf. Commun. Technol. Res. ICTRC 2015*, no. April, pp. 222–225, 2015.

5 Y. Yu, S. L. Kang, and H. Qiu, "A new network intrusion detection algorithm: DA-ROS-ELM," *IEEJ Trans. Electr. Electron. Eng.*, vol. 13, no. 4, pp. 602–612, 2018.

6 Z. Xu, M. Yao, Z. Wu, and W. Dai, "Incremental regularized extreme learning machine and it's enhancement," *Neurocomputing*, vol. 174, pp. 134–142, 2016.

7 C. Lu, H. Ke, G. Zhang, Y. Mei, and H. Xu, "An improved weighted extreme learning machine for imbalanced data classification," *Memetic Comput.*, vol. 11, no. 1, pp. 27–34, 2019.

8 A. Saxena, M. Ieee, K. Goebel, D. Simon, and N. Eklund, “Damage Propagation Modeling for Aircraft Engine Prognostics,” *Response*, 2008.
9 J. J. L. F. Dean, J. de Castro, *User’s Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)*. Cleveland, OH, USA,: NASA Center for AeroSpace Information:, 2007.
10 BERGHOUT Tarek, Mouss Leila Hayet, Kadri Ouahab, “RUL prediction (C-MAPSS dataset),” 2019. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/73629-rul-prediction-c-mapss-dataset.
11 G. Bin Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: A new learning scheme of feedforward neural networks,” *IEEE Int. Conf. Neural Networks - Conf. Proc.*, vol. 2, no. August, pp. 985–990, 2004.
12 M. N. Alam, “Codes in MATLAB for Particle Swarm Optimization Codes in MATLAB for Particle Swarm Optimization,” no. March, 2016.
13 C. Zheng *et al.*, “A Data-driven Approach for Remaining Useful Life Prediction of Aircraft Engines,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2018-Novem, pp. 184–189, 2018.



Tarek BERGHOUT was born in 1991 in RAHBAT-Algeria, he studied in BATNA University (Algeria), and he has a Master degree in industrial engineering and manufacturing (2015). Currently he is a PhD Researcher and codes writer specialized in industrial prognosis based on Machine Learning tools.
Interests: Extreme Learning Machine, dynamic data compression with deep ANNs, “data-driven” prediction based Deep ANNs, Time varying data challenges, linear approximation and dynamic, programming, big data and Deep Learning.