Neurocomputing 146 (2014) 75–82

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A study on residence error of training an extreme learning machine and its application to evolutionary algorithms

Ai-Min Fu [a], Xi-Zhao Wang [b], Yu-Lin He [b,*], Lai-Sheng Wang [a,**]

[a] College of Science, China Agricultural University, Beijing 100083, China
[b] Key Laboratory of Machine Learning and Computational Intelligence in Hebei Province, School of Mathematics and Computer Science, Hebei University, Baoding 071002, China

ARTICLE INFO

Article history:
Received 23 September 2013
Received in revised form
9 April 2014
Accepted 14 April 2014
Available online 24 July 2014

Keywords:
Extreme learning machine
Genetic algorithm
Rank of matrix
Residence error
Solution stability

ABSTRACT

This paper delivers a study on the change of rank of input matrix in Extreme Learning Machine (ELM) and the relationship between the rank of input matrix and the residence error of training an ELM. From the viewpoint of data analysis, the study reveals why ELM has a decreasing residence error with the increase of number of nodes in hidden layer and what role the Sigmoid function plays in increasing the rank of input matrix. Furthermore the relationship between the stability of solutions and the rank of output matrix is also discussed. An application of residence error to genetic algorithms of minimizing L_1-norm ELM is given.

1. Introduction

Extreme learning machines (ELMs) proposed in [14,15] are a type of single hidden layer feed-forward neural networks (SLFNs) in which the weights between input layer and hidden layer are chosen randomly while the weights between hidden layer and output layer are obtained by solving a system of linear matrix equations. ELMs adopt the sum of squared losses on the training error as the objective function, and then turn training of output weights into a regularized least square problem. It has been shown that SLFNs only with randomly generated input weights and tunable output weights can maintain their universal approximation ability [10,11,29]. In comparison with gradient-descent based algorithms, ELMs have much more efficient training and usually lead to better generalization performance [21,25,27].

One can find considerable references [2,12,13,18,24–29] in recent decade regarding the ELM study. We are now interested in ELM's training residence error and approximation capability, and a very brief literature review is given as follows. Hornik in [6] proved that, if the activation function is continuous, bounded, and non-constant, then continuous mappings can be approximated by SLFNs with additive hidden nodes over compact input sets. Leshno et al. in [16] improved the result of [6] by proving that SLFNs with additive hidden nodes and with a non-polynomial activation function can approximate any continuous target functions [16]. Huang and Babri [7] show that an SLFN with at most N hidden nodes and with almost any nonlinear activation function can learn N distinct observations with zero error, where N is the number of training samples. Furthermore, Huang et al. [8–10] recently proposed a series of learning algorithms referred to as incremental extreme learning machines (I-ELMs) where the number of hidden layer nodes are gradually added and showed that such I-ELMs can converge to any continuous function as long as the hidden activation functions are nonlinear piecewise continuous. Following [9], Feng et al. [4] proposed an error minimized extreme learning machine (EM-ELM), which can add random hidden nodes to SLFNs one by one or group by group (with varying group size). During the growth of the networks, the output weights are updated incrementally. The convergence of this approach is proved.

Based on the result of [1] in which the authors pointed out that for the feed-forward neural networks, the smaller the norm of weights and training error is, the better generalization performance the networks tend to have, all ELM algorithms tend to find the minimum norm least square solution such that a smaller training error can be achieved. This study focuses on the change of rank of input matrix in ELM and the relationship between the rank of input matrix and the residence error of training an ELM. It theoretically confirms that the increase of input dimension given by random weights and the increase of rank of middle matrix

* Corresponding author. Tel.: +86 18531315747.
** Corresponding author.
E-mail addresses: yulinhe@ieee.org (Y.-L. He), wanglaish@126.com (L.-S. Wang).

http://dx.doi.org/10.1016/j.neucom.2014.04.067
0925-2312/

induced by Sigmoid transformation play the crucial role in the entire process of training an ELM.

All existing references indicate that ELM is a useful and efficient technique for supervised learning. But there is no article yet to clearly explain why the ELM can effectively work well with a simple structure and fast training. This paper makes an attempt to give an explanation from the angle of relationship between ELM's training residence error and the rank of input matrix.

The rest of this paper is organized as follows. Section 2 lists a brief review on the approximation ability and error analysis of ELMs. Section 3 investigates the increase of dimension for input matrix and the relationship of rank between input matrix and middle matrix, and Section 4 studies the impact of Sigmoid transformation on the increase of rank of output matrix. Section 5 studies the estimation of residence error and stability of solution. Then, an application of residence error to genetic algorithms of minimizing L_1-norm ELM is given in Section 6. Section 7 of this paper presents our conclusion.

2. Extreme learning machine

Usually an ELM means a three layer neural network in which the weights between input layer and hidden layer are randomly selected and the weights between hidden layer and output layer are determined by solving a generalized system of linear equations (i.e., by computing the pseudo inverse of a matrix). Fig. 1 depicts the basic structure of an ELM in which we suppose that the input layer has n nodes, the hidden layer has m nodes, and the output layer have has only one node.

We now analyze the training process of an ELM. The training task is to determine the connection weights r_{ij} and β_j $(i=1,2,\cdots,n;\ j=1,2,\cdots,m)$. Since the weights r_{ij} $(i=1,2,\cdots,n;\ j=1,2,\cdots,m)$ are randomly selected, the training task is reduced to determine β_j $(j=1,2,\cdots,m)$ only. Suppose that the set of training data contains N examples which can be expressed as an input matrix A (with N rows and n columns) and a N-dimensional output vector b, respectively denoted by

$$A_{N\times n}=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{N_1} & a_{N_2} & \cdots & a_{N_n} \end{pmatrix} \text{ and } b_{N\times 1}=\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}.$$

The weights between the input layer and hidden layer are expressed as a matrix with n rows and m columns, i.e., $R=(r_{ij})_{n\times m}$, and the weights between the hidden layer and output layer are denoted as an m-dimensional vector, i.e., $\beta=(\beta_1,\beta_2,\cdots,\beta_m)^T$. Let

$$S_{N\times m}\triangleq \underset{N\times n}{A}\ \underset{n\times m}{R}=(s_{ij})_{N\times m} \text{ and } \underset{N\times m}{H}\triangleq (f(S_{ij}))_{N\times m}=(h_{ij})_{N\times m},$$

where $f(x)=(1/1+e^{-x})$ denotes the Sigmoid function. Then the training task of the ELM is transferred to solve the following system of linear equations $H\beta=b$, which is equivalent to the following optimization problem:

$$\min_{\beta\in R^m}\| \underset{N\times 1}{b}-\underset{N\times m}{H}\ \underset{m\times 1}{\beta}\|^2. \tag{1}$$

Obviously the solution of Eq. (1) is not unique in a general case. From the viewpoint of regularization, Huang et al. [10,13,15] suggested to use the minimum-norm minimum least square solution as the final one:

$$\min\left(\min_{\beta\in R^m}\| \underset{N\times 1}{b}-\underset{N\times m}{H}\ \underset{m\times 1}{\beta}\|^2\right). \tag{2}$$

It is easy to see that the solutions of Eqs. (1) and (2) can be respectively expressed as

$$\beta=H^- b \text{ and } \beta=H^+ b,$$

where H^- denotes any generalized inverse matrix H while H^+ denotes the plus-generalized inverse this is unique for an matrix H.

We divided the above-mentioned training process of an ELM as three steps: (1) dimension increase for input matrix; (2) rank increase for output matrix; and (3) solving a system of linear equations with full rank matrix of coefficients. The three steps are depicted in Fig. 2.

Step 1 shows a process of increasing dimension of input matrix A since in almost every case of ELM applications the number of hidden nodes is much bigger than the number of input nodes. Ref. [24] discussed the impact of increasing dimension of input matrix and pointed out that without the dimension increase the ELM will not obtain a good generalization and approximation ability. In fact, the central supporting theorem of EML algorithms, given by Zhang et al. in [29], stated a process of approximation with the increase of number of hidden nodes.

Step 2 gives a process of increasing rank of input matrix. Although in step 1 the input matrix A (with N rows and n columns) becomes S (with N rows and m columns) through the multiplication to a random weight matrix R and m is bigger than n, the rank of matrix S is less than or equal to the rank of matrix A. It is because the step 1 is only a linear transformation (the simple proof remains in next section). The essence of step 2 is a nonlinear transformation. ELM uses a Sigmoid function which usually plays a role of transforming from a waning rank matrix S to a full rank matrix H.

Step 3 means to solve a system of linear equations. If the output layer has more than one node then it is a system of linear matrix equations. It is well known that the criterion of least square is frequently used to solve the system. It is evaluated by the approximation error (i.e., the residence error). Here we are mainly interested in the relationship between the approximation error and the rank of coefficient matrix.

To be convenient for our following discussions, we summarize the used symbols or notations as follows:

A—input matrix;
R—random weight matrix;
β—weight vector to be solved;
S—middle matrix;
H—output matrix;
H^+—solution matrix;
b—expected output vector.

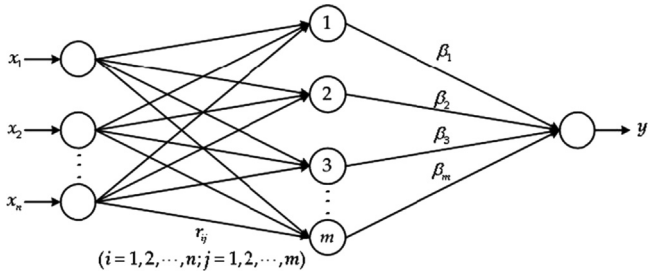

Fig. 1. A simple ELM structure.

3. Increase of dimension for input matrix

This section has two aims. One is to make clear the impact of dimension increase (from A to S) on the solution of $H\beta=b$, the