



# The conjugate gradient optimized regularized extreme learning machine for estimating state of charge

Meng Jiao<sup>1</sup> · Yan Yang<sup>1</sup> · Dongqing Wang<sup>1</sup> · Peng Gong<sup>1</sup>

Received: 31 May 2021 / Revised: 22 June 2021 / Accepted: 24 June 2021 / Published online: 31 August 2021  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

## Abstract

In this paper, a conjugate gradient optimized regularized extreme learning machine (CG-RELM) is investigated to estimate the state of charge (SOC) of lithium batteries. The CG algorithm naturally avoids calculation load of the inverse matrix in the least square algorithm in updating output weights of the RELM. In addition, the weights adjustment directions in the CG algorithm at different points are conjugate orthogonal to each other, guaranteeing the high convergence speed of the CG-RELM algorithm. The simulation results verify that the investigated algorithm can effectively estimate the battery SOC under the dynamic stress test (DST) data set. The CG-RELM shows faster convergence speed when compared with BP neural network, higher estimation accuracy when the number of hidden layer neurons increases, and high robustness when applied to the data set with noise.

**Keyword** Conjugate gradient · Regularized extreme learning machine · State of charge

## Introduction

With the development of power generation technology and energy storage technology, the electric vehicles (EVs) have become a new direction for the development of the automotive industry. Lithium batteries are ideal power sources of EVs due to their excellent performance [1, 2]. An essential link between batteries and EVs is the battery management system (BMS) [3, 4]. However, the most important parameter in BMS, the State of Charge (SOC), cannot be obtained by direct measurement [5]. Therefore, accurately estimating the SOC of lithium batteries has become a very important issue.

## The SOC estimation methods

A number of SOC estimation methods have been proposed by researchers, which can be divided into the following categories.

### 1) *The traditional methods*

✉ Dongqing Wang  
dqwang64@qdu.edu.cn

The traditional methods contain the Coulomb counting (CC) method [6] and the open-circuit voltage (OCV) method [7]. The CC method obtains the SOC by integrating the battery current over time during charging or discharging. However, its estimation performance depends on whether there is an exact initial SOC and a highly accurate sensor. If not, cumulative errors may occur, which will affect the final estimation result [7]. The OCV method obtains the SOC by looking up the SOC-OCV curve. But it requires a long time to keep the battery in an open-circuit state before measuring its OCV. This makes the method unable to meet the requirements of real-time estimation [8]. Furthermore, if the SOC-OCV curve obtained in advance is not accurate enough, the estimation performance of the OCV method may be counterproductive.

### 2) *The Kalman filter-based methods*

The Kalman filter-based methods estimate the battery SOC by combining an electrical circuit model (ECM) [9, 10] with an appropriate Kalman filter (KF), e.g., the extended Kalman filter (EKF) [11–13], the unscented Kalman filter (UKF) [14], the dual Kalman filter (DKF) [15], and the cubature Kalman filter (CKF) [16, 17], as well as other filters [18–20]. These strategies typically require accurate battery models to perform SOC estimation at varying conditions. Practically, there are a large number of parameters which are difficult to be identified in the battery model. In addition, only when the noise is zero

<sup>1</sup> College of Electrical Engineering, Qingdao University, 308 Ningxia Road, Qingdao 266071, China

mean (also called the Gaussian noise), the estimation results of the Kalman filter-based methods are accurate [21]. However, in a wide variety of applications, the noise is non-Gaussian.

### 3) The neural network-based methods

The neural network is an efficient recognition algorithm widely used in fault location [22, 23], classification [24], pattern recognition [25], and other fields. When applied to SOC estimation, the neural network-based methods can infer the relationship between the battery current, voltage, other variables, and SOC according to a large amount of training data [26, 27]. These methods realize the optimization of weights and biases through iterative training. The commonly used optimization algorithm is the gradient descent (GD) algorithm [28, 29]. However, during the training process based on the GD algorithm, when the weights of some layers in the neural network change significantly, the weights of other layers usually remain unchanged and only change during further iterations. In other words, the GD algorithm has inherent instability. Therefore, it will take a long time to update all the weights and biases of the network to an ideal state.

### The extreme learning machine

The extreme learning machine (ELM) was proposed by Huang et al. in 2004 [30]. It is a special single-hidden layer feedforward neural network (SLFN). In an ELM, the input weights and hidden layer biases are set randomly and remain unchanged during the training process. The output weights are calculated according to the batch data least square algorithm [31]. Compared with the GD algorithm-based neural network, the training steps of the ELM are simpler, which makes its training speed faster. However, a potential disadvantage of the ELM is that the improvement of its learning accuracy is guaranteed by increasing the number of neurons in the hidden layer [32]. When dealing with complex problems, the structure of the ELM is often too large due to numerous neurons; then, over-fitting may occur, which will reduce the generalization ability of the algorithm [33].

Huang et al. introduced the regularization term into the objective function of the standard ELM and then got the regularized extreme learning machine (RELM) [34]. Correspondingly, during the training of the RELM by using the least square algorithm, the regularization coefficient is included in the Moore-Penrose generalized inverse matrix of the hidden output. In this way, both the empirical risk and the structural risk of the model can be reduced. Therefore, the generalization ability of the model can be improved, and the over-fitting can be effectively prevented.

With its excellent learning performance, many RELM-based methods have been proposed and have been widely used in various fields. For example, an online sequential regularized extreme learning machine (OS-RELM) was proposed by Cosmo et al. for

single image super resolution and achieved comparable or better results than some state-of-the-art methods [35]. Gumaei et al. used a hybrid feature extraction method of a RELM for brain tumor classification with superior classification results of brain images [36]. Weng et al. applied a BIC criterion and genetic algorithm-based RELM in iron ore price forecasting [37].

For the ELM, its inverse matrix, which has a high computational complexity, needs to be calculated during the training process of the standard ELM and its variants [38]. Furthermore, a large number of intermediate matrices need to be stored in the process of calculating the inverse matrix, which leads to a large occupation of computer memory.

### Contributions

In this paper, a conjugate gradient optimized regularized extreme learning machine (CG-RELM) is proposed to estimate battery SOC. The main contributions of this study reflect in the following aspects:

- 1) For a lithium battery, a CG-RELM model is established for SOC estimation with the measured voltage and current as the inputs, and the estimated SOC as the output of the model.
- 2) In the CG-RELM, the conjugate gradient (CG) algorithm is used to calculate the output weights. This not only avoids calculating the inverse matrix but guarantees the high convergence speed of the algorithm.
- 3) In simulation, the dynamic stress test (DST) data set is used to train the CG-RELM, and the simulation results verify that the proposed algorithm can effectively estimate the battery SOC.

This paper is set as follows. Section 2 illustrates the principle of the RELM. Section 3 introduces the principle of the CG algorithm. Section 4 investigates the CG algorithm optimized RELM for battery SOC estimation. Section 5 constructs the experimental platform, performs simulation, and analyzes the results of the CG-RELM model. The conclusion is given in Section 6.

### The regularized extreme learning machine

The regularized extreme learning machine (RELM) is a kind of feedforward neural network with a single-hidden layer.

Notations:

- $M$ ,  $L$ , and  $N$  indicate the number of neurons in the input layer, the hidden layer, and the output layer, respectively.
- $X$  and  $Y$  indicate the input matrix and the target output matrix of the network, respectively.
- $O$  indicates the output matrix of the network.
- $\sigma(\cdot)$  indicates the Sigmoid activation function of the hidden layer.

- $W^1$ ,  $b$ , and  $W^2$  indicate the input weight matrix, the bias vector, and the output weight matrix of the network.

Based on the above notation explanations,  $X$  and  $Y$  can be expressed as follows:

$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_M^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_M^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(S)} & x_2^{(S)} & \cdots & x_M^{(S)} \end{bmatrix} = \begin{bmatrix} X^{(1)} \\ X^{(2)} \\ \vdots \\ X^{(S)} \end{bmatrix}_{S \times M}$$

$$Y = \begin{bmatrix} y_1^{(1)} & y_2^{(1)} & \cdots & y_N^{(1)} \\ y_1^{(2)} & y_2^{(2)} & \cdots & y_N^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ y_1^{(S)} & y_2^{(S)} & \cdots & y_N^{(S)} \end{bmatrix} = \begin{bmatrix} Y^{(1)} \\ Y^{(2)} \\ \vdots \\ Y^{(S)} \end{bmatrix}_{S \times N}$$

where  $S$  indicates the sample size.

$W^1$ ,  $b$ , and  $W^2$  can be expressed as follows:

$$W^1 = \begin{bmatrix} w_{11}^1 & w_{12}^1 & \cdots & w_{1L}^1 \\ w_{21}^1 & w_{22}^1 & \cdots & w_{2L}^1 \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1}^1 & w_{M2}^1 & \cdots & w_{ML}^1 \end{bmatrix} = [W_1^1 W_2^1 \cdots W_L^1]_{M \times L}$$

$$b = [b_1 b_2 \cdots b_L]_{L \times 1}^T$$

$$W^2 = \begin{bmatrix} w_{11}^2 & w_{12}^2 & \cdots & w_{1N}^2 \\ w_{21}^2 & w_{22}^2 & \cdots & w_{2N}^2 \\ \vdots & \vdots & \ddots & \vdots \\ w_{L1}^2 & w_{L2}^2 & \cdots & w_{LN}^2 \end{bmatrix} = [W_1^2 W_2^2 \cdots W_L^2]_{L \times N}$$

$$b = [b_1 \quad b_2 \quad \cdots \quad b_L]_{L \times 1}^T$$

$$W^2 = \begin{bmatrix} w_{11}^2 & w_{12}^2 & \cdots & w_{1N}^2 & w_{L1}^2 & w_{L2}^2 & \cdots & w_{LN}^2 \\ w_{21}^2 & w_{22}^2 & \cdots & w_{2N}^2 & & & & \\ \vdots & \vdots & \ddots & \vdots & & & & \\ w_{L1}^2 & w_{L2}^2 & \cdots & w_{LN}^2 & & & & \end{bmatrix} = [W_1^2 \quad W_2^2 \quad \cdots \quad W_L^2]_{L \times N}$$

Then the output matrix of the hidden layer can be calculated as:

$$H = \begin{bmatrix} \sigma(X^{(1)} \cdot W_1^1 + b_1) & \cdots & \sigma(X^{(1)} \cdot W_L^1 + b_L) \\ \vdots & \ddots & \vdots \\ \sigma(X^{(S)} \cdot W_1^1 + b_1) & \cdots & \sigma(X^{(S)} \cdot W_L^1 + b_L) \end{bmatrix}_{S \times L} \tag{1}$$

Finally, the output matrix of the network can be obtained:

$$O = HW^2. \tag{2}$$

According to the RELM algorithm, the input weight matrix ( $W^1$ ) and bias vector ( $b$ ) are set randomly and stay unchanged during the training process. Therefore, the task of the RELM is to find an estimate  $W^{2,*}$  that can minimize the error between the output ( $O$ ) and the target output ( $Y$ ) of the network, which can be expressed as:

$$W^{2,*} = \min_{W^2} \|O - Y\| = \min_{W^2} \|HW^2 - Y\|. \tag{3}$$

The traditional ELM algorithm calculates  $W^{2,*}$  by using the batch data least square algorithm, as shown below:

$$W^{2,*} = H^+ Y = (H^T H)^{-1} H^T Y. \tag{4}$$

where  $H^+ = (H^T H)^{-1} H^T$  indicates the generalized Moore-Penrose inverse matrix of  $H$ .

However, when dealing with complex problems, in order to meet the requirement of accuracy, the hidden layer neurons of the ELM are usually set too much. This may lead to overfitting, thereby reducing the generalization ability of the network. Moreover, when  $H$  is not a column full-rank matrix, the result of the determinant of  $H^T H$  is 0. Then, an error may occur when calculating  $W^{2,*}$  according to (4).

To solve the above problems, the RELM adds a positive constant  $\lambda$  (called the regularization coefficient) to each element on the diagonal of  $H^T H$ . Then,  $W^{2,*}$  can be calculated as follows:

$$W^{2,*} = (H^T H + \lambda I)^{-1} H^T Y. \tag{5}$$

where  $I$  is an identity matrix.

## The conjugate gradient algorithm

### The related work

**Definition 3.1** An  $n \times n$  matrix  $A$  is positive definite if  $z^T A z > 0$  for every  $n$ -dimensional column vector  $z \neq 0$ , where  $z^T$  is the transpose of vector  $z$ .

**Definition 3.2** An  $n \times n$  matrix  $A$  is positive semi-definite if  $z^T A z \geq 0$  for every  $n$ -dimensional column vector  $z \neq 0$ , where  $z^T$  is the transpose of vector  $z$ .

**Definition 3.3** Two vectors,  $r_1$  and  $r_2$ , are conjugate with respect to  $A$  (called  $A$ -conjugate) if  $r_1^T A r_2 = 0$ , where  $A$  is an  $n \times n$  symmetric positive definite matrix and  $r_1, r_2 \in \mathbb{R}^n$ .

Also,  $r_1, r_2, \dots, r_k \in \mathbb{R}^n$  are a set of conjugate directions of  $A$  if  $r_i^T A r_j = 0, i \neq j, i, j = 1, 2, \dots, k$ .

**Remark 3.1** The notion of conjugacy is in some ways a generalization of orthogonality. When  $A$  is the identity matrix  $I$ ,  $A$ -conjugate means  $A$ -orthogonal.

**Theorem 3.1** Suppose there is a linear system  $Ax = B$ , where  $A$  is the coefficient matrix,  $x$  is the solution vector, and  $B$  is the constant vector. If  $A$  is a symmetric positive definite matrix, then, the solution of  $Ax = B$  is also the minimal point (denoted as  $x^*$ ) of the function  $f(x)$  shown below:

$$f(x) = \frac{1}{2}x^T Ax - B^T x.$$

**Theorem 3.2** Suppose  $r_1, r_2, \dots, r_n \in \mathbb{R}^n$  are a set of  $A$ -conjugate vectors, where  $A$  is an  $n \times n$  symmetric positive definite matrix. Take any  $x_1 \in \mathbb{R}^n$  as the initial point, a series of new points can be obtained by performing one-dimensional search in the direction of  $r_1, r_2, \dots, r_n$ . The minimal point of the function  $f(x)$  in Theorem 3.1 can be obtained by iterating at most  $n$  times—namely, the conjugate direction algorithm.

According to Theorems 3.1–3.2, if there is a set of  $A$ -conjugate direction vectors  $r_1, r_2, \dots, r_n \in \mathbb{R}^n$ , the solution of  $Ax = B$  can be obtained by the conjugate direction algorithm.

### The principle of the conjugate gradient algorithm

The conjugate gradient (CG) algorithm [39], a kind of conjugate direction algorithm, generates  $A$ -conjugate direction vectors by using the gradient vectors of  $f(x)$ . Then, the minimal point of  $f(x)$  can be calculated, which is also the solution of the linear equation  $Ax = B$ .

**Remark 3.2** The gradient of the function  $f(x)$  can be obtained by calculating the partial derivative of  $f(x)$  against  $x$ .

According to Theorem 3.1 and Remark 3.2, we have:

$$g_k = \left. \frac{\partial f}{\partial x} \right|_{x=x_k} = Ax_k - B, \tag{6}$$

$$Ax^* - B = 0. \tag{7}$$

According to the CG algorithm, the direction vector at the  $k$ th iteration ( $r_k$ ) is obtained from that at the previous iteration ( $r_{k-1}$ ) and the gradient vector at the current iteration ( $g_k$ ), as shown below:

$$r_k = -g_k + \beta_k r_{k-1}. \tag{8}$$

where  $\beta_k$  is the correction coefficient.

According to Definition 3.3 and (8),  $\beta_k$  in the CG algorithm need meet:

$$r_k^T Ar_{k-1} = -g_k^T Ar_{k-1} + \beta_k r_{k-1}^T Ar_{k-1} = 0. \tag{9}$$

Then, we have the calculation equation of  $\beta_k$ :

$$\beta_k = \frac{g_k^T Ar_{k-1}}{r_{k-1}^T Ar_{k-1}}. \tag{10}$$

**Definition 3.4** Define the learning rate at the  $k$ th iteration as  $\alpha_k$ . Then, the parameter updating equation of the CG algorithm is:

$$x_{k+1} = x_k + \alpha_k r_k. \tag{11}$$

**Definition 3.5** Define  $e_k = x^* - x_k$  as the error vector at the  $k$ th iteration, and  $r_{k-1}$  is the direction vector at the previous iteration. The accurate one-dimensional search requires that  $e_k$  and  $r_{k-1}$  be conjugate with respect to  $A$ ; that is,  $r_{k-1}^T A e_k = 0$ .

Then, we can get:

$$\begin{aligned} r_k^T A e_{k+1} &= r_k^T A (x^* - x_{k+1}) \\ &= r_k^T A (x^* - x_k + x_k - x_{k+1}) \\ &= r_k^T A (e_k - \alpha_k r_k) \\ &= r_k^T A e_k - \alpha_k r_k^T A r_k \\ &= 0. \end{aligned} \tag{12}$$

Further, the calculation of  $\alpha_k$  can be obtained:

$$\alpha_k = \frac{r_k^T A e_k}{r_k^T A r_k}. \tag{13}$$

Substituting (6) and (7) into (13), the simplified expression of  $\alpha_k$  can be obtained as follows:

$$\alpha_k = \frac{r_k^T A (x^* - x_k)}{r_k^T A r_k} = \frac{r_k^T (B - Ax_k)}{r_k^T A r_k} = -\frac{r_k^T g_k}{r_k^T A r_k}. \tag{14}$$

The expression of  $\beta_k$  in (10) can also be simplified, and there are many equations for simplification, such as the Polak-Ribière-Polyak (PRP) equation [40], the Fletcher-Reeves (FR) equation [41], the conjugate descent (CD) equation [42], the Liu-Storey (LS) equation [43], and the Dai-Yuan (DY) equation [44].

In this paper, the FR equation is selected. The advantage of the FR equation is that only the gradient vectors are used when calculating  $\beta_k$ , and the coefficient matrix and the direction vectors are not needed.

The derivation of the FR equation is as follows:

According to Definition 3.5 and (11), when  $i < j$ , the error vector at the  $j$ th iteration can be expressed as follows:

$$\begin{aligned} e_j &= (x_{i+1} + e_{i+1}) - x_j \\ &= e_{i+1} - (x_j - x_{i+1}) \\ &= e_{i+1} - \sum_{k=i+1}^{j-1} \alpha_k r_k. \end{aligned} \tag{15}$$

Then, according to (6), (7), and (15), we have:

$$\begin{aligned} r_i^T g_j &= r_i^T (Ax_j - B) \\ &= r_i^T (Ax_j - Ax^*) \\ &= -r_i^T A e_j \\ &= -r_i^T A \left( e_{i+1} - \sum_{k=i+1}^{j-1} \alpha_k r_k \right). \end{aligned} \tag{16}$$

According to Definition 3.3 and Definition 3.5, we have:  $r_i^T A e_{i+1} = 0$  and  $r_i^T A r_k = 0, k = i + 1, \dots, j - 1$ . Then, we get:

$$r_i^T g_j = 0, i < j. \tag{17}$$

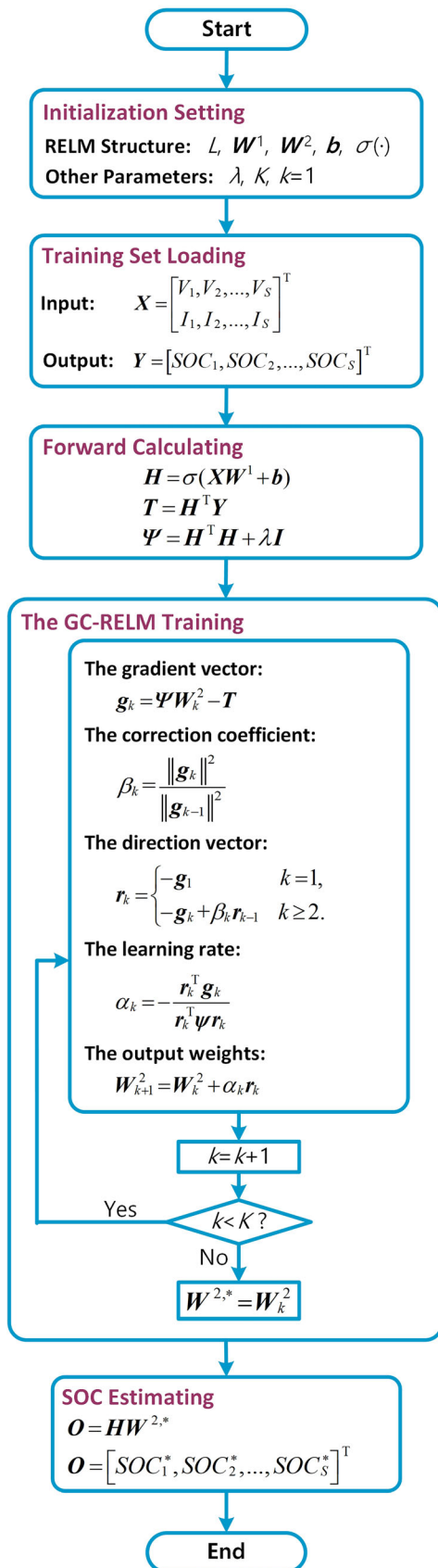


Fig. 1 The training steps of the CG-RELM

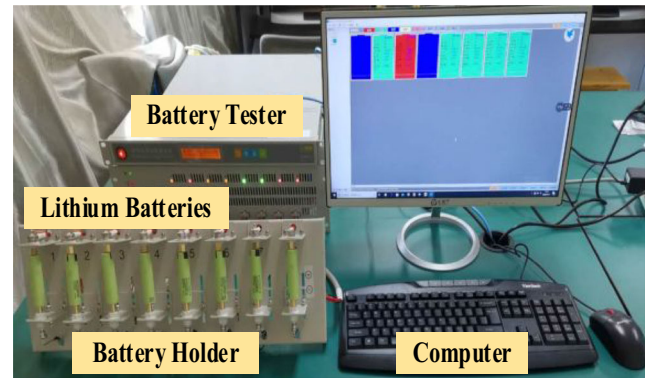


Fig. 2 The battery test platform

Substituting (11) into (6) gives:

$$\begin{aligned}
 g_k &= A(x_{k-1} + \alpha_{k-1}r_{k-1}) - B \\
 &= Ax_{k-1} - B + \alpha_{k-1}Ar_{k-1} \\
 &= g_{k-1} + \alpha_{k-1}Ar_{k-1}.
 \end{aligned}
 \tag{18}$$

Then, we have:

$$Ar_{k-1} = \frac{g_k - g_{k-1}}{\alpha_{k-1}}.
 \tag{19}$$

Substituting (19) into (10) gives:

$$\beta_k = \frac{g_k^T g_k - g_{k-1}^T g_{k-1}}{r_{k-1}^T g_k - r_{k-1}^T g_{k-1}}.
 \tag{20}$$

According to (8) and (17), we have:

$$\begin{aligned}
 r_{k-1}^T g_k &= (-g_{k-1}^T + \beta_{k-1}r_{k-2}^T)g_k \\
 &= -g_{k-1}^T g_k + \beta_{k-1}r_{k-2}^T g_k \\
 &= -g_{k-1}^T g_k \\
 &= 0.
 \end{aligned}
 \tag{21}$$

Then, we get:

$$g_{k-1}^T g_k = 0.
 \tag{22}$$

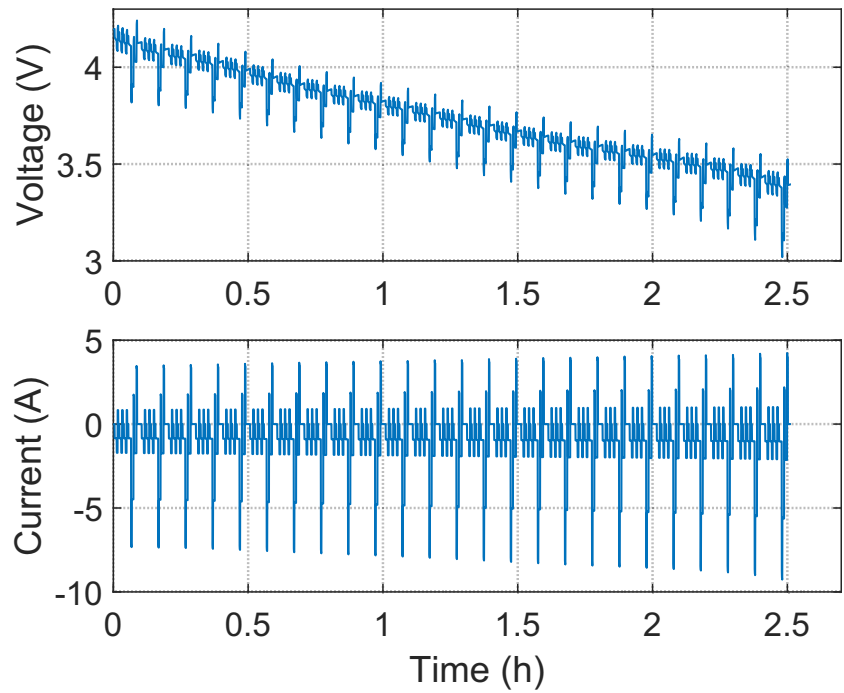
Therefore, we have:

$$\begin{aligned}
 r_{k-1}^T g_{k-1} &= (-g_{k-1}^T + \beta_{k-1}r_{k-2}^T)g_{k-1} \\
 &= -g_{k-1}^T g_{k-1} + \beta_{k-1}r_{k-2}^T g_{k-1} \\
 &= -g_{k-1}^T g_{k-1}.
 \end{aligned}
 \tag{23}$$

Table 1 Indexes of the NCR18650PF lithium battery

Capacity	2900 mAh
Operating voltage	2.7V–4.2V
Rated voltage	3.7V
Rated charge current	1.45 A (0.5C)
Max. discharge current	8.7 A (3C)

**Fig. 3** The voltage and current of the DST data set



Substituting (21)–(23) into (20), we obtain the FR equation:

$$\beta_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} = \frac{\|\mathbf{g}_k\|^2}{\|\mathbf{g}_{k-1}\|^2}. \tag{24}$$

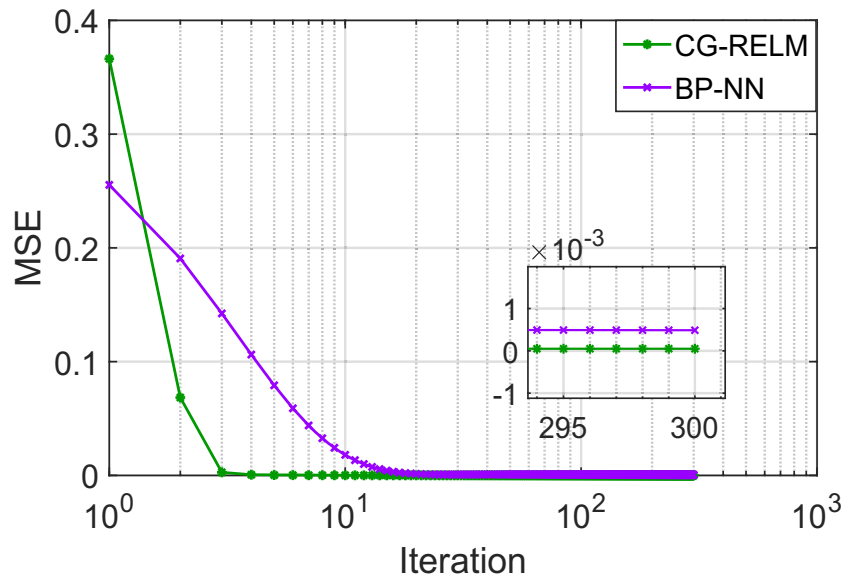
$$\mathbf{r}_k = \begin{cases} -\mathbf{g}_1 & k = 1, \\ -\mathbf{g}_k + \frac{(\mathbf{g}_k)^2}{(\mathbf{g}_{k-1})^2} \mathbf{r}_{k-1} & k \geq 2. \end{cases} \tag{25}$$

Choose the opposite gradient direction as the initial direction. Then, the CG algorithm can generate a set of  $\mathbf{A}$ -conjugate direction vectors  $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n \in \mathbb{R}^n$  according to the following equation:

**The CG-RELM algorithm for SOC estimation**

The conjugate gradient optimized regularized extreme learning machine (CG-RELM) is used for battery SOC estimation

**Fig. 4** The MSE of the CG-RELM and BP-NN during training





**Table 2** The performance of different models

Models	Indexes	RMSE	MAE	R <sup>2</sup>
CG-RELM	Training set	0.0071	0.0051	0.9994
	Test set	0.0075	0.0053	0.9993
BP-NN	Training set	0.0479	0.0437	0.9640
	Test set	0.0475	0.0433	0.9639

by taking the measured voltage  $V_i$  and current  $I_i$  ( $i=1,2,\dots,S$ ) as inputs and the estimated  $SOC_i$  as the output. Therefore, the number of neurons in the input layer and the output layer is 2 and 1, respectively, and we have:

$$X = \begin{bmatrix} V_1, V_2, \dots, V_s \\ I_1, I_2, \dots, I_s \end{bmatrix}^T, Y = [SOC_1, SOC_2, \dots, SOC_s]^T.$$

Let  $\Psi = H^T H + \lambda I$  and  $T = H^T Y$ ; then, Eq. (5) can be transformed into:

$$\Psi W^{2,*} = T. \tag{26}$$

According to Definitions 3.2–3.3,  $\Psi$  is a symmetric positive definite matrix. The linear equation  $\Psi W^{2,*} = T$  is same as  $Ax = B$ . Thus, according to Theorems 3.1–3.2, its solution can copy the results of  $Ax = B$  by the CG algorithm as:

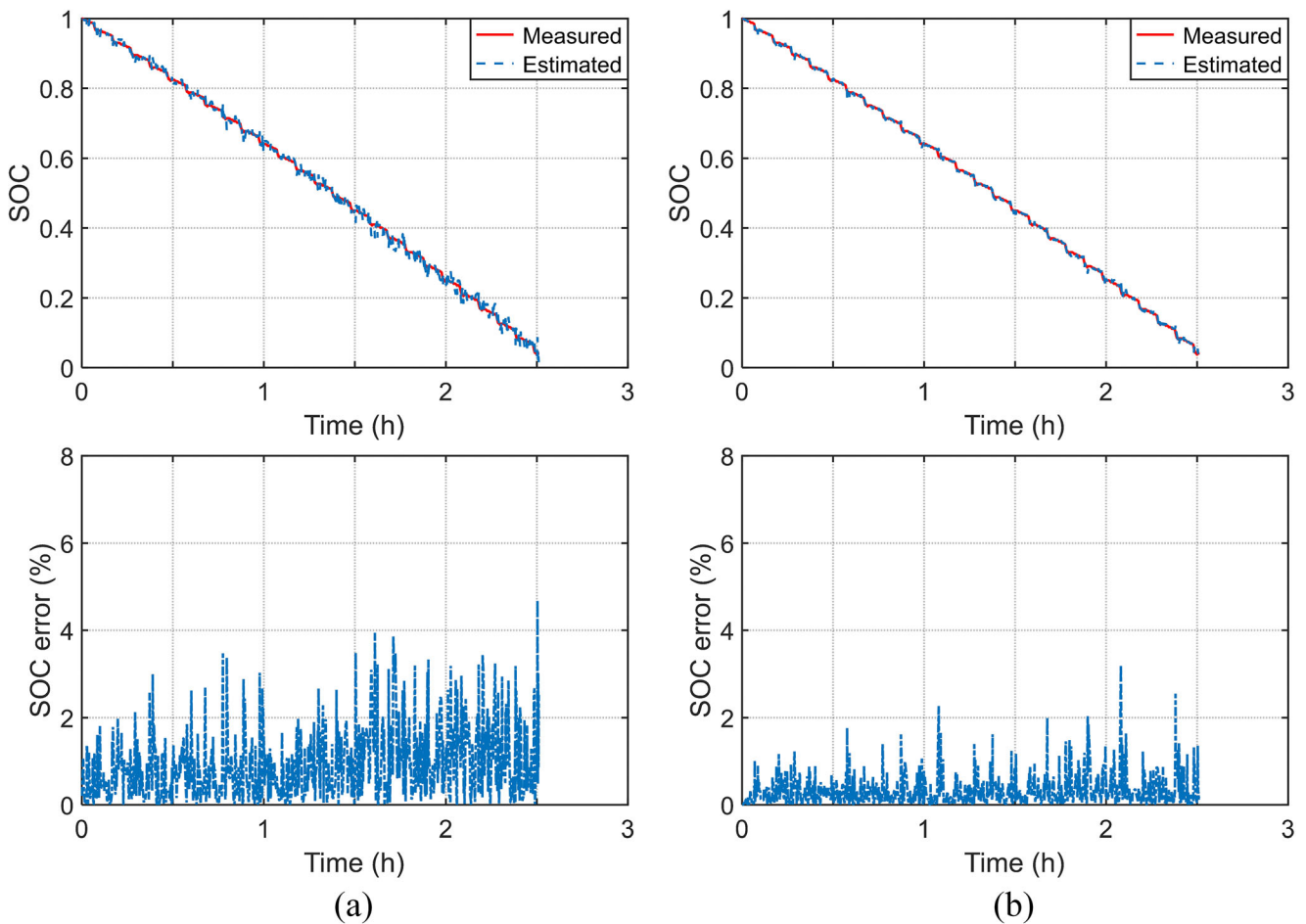
$$W^{2,*} = W_{k+1}^2 = W_k^2 + \alpha_k r_k. \tag{27}$$

The training steps of the CG-RELM algorithm are shown in Fig. 1.

## Experiment and simulation

### Data sampling and preprocessing

As shown in Fig. 2, there is a battery test platform which consists of a battery tester (NEWARE CT-4008-5V12A-TB), a battery holder, NCR18650PF (2900mAh) lithium batteries, and a computer. The battery indexes are indicated in Table 1. Perform discharge test on this battery test platform to sample data, and then, get the dynamic stress test (DST) data set, which is shown in Fig. 3.



**Fig. 5** The estimated SOC and errors of the CG-RELM with different neuron numbers. **a** The estimated SOC and errors when  $L = 200$  **b** The estimated SOC and errors when  $L = 600$

**Table 3** The performance of the CG-RELM with different neuron numbers

$L$	Indexes	RMSE	MAE	$R^2$
200	Training set	0.0119	0.0091	0.9981
	Test set	0.0117	0.0091	0.9982
600	Training set	0.0043	0.0030	0.9998
	Test set	0.0048	0.0032	0.9997

First, we adopt the exponential moving average method to process the original voltage and current data sampled during the discharge process. The calculation formula is as follows:

$$\begin{cases} y'_k = y_k & k=1, \\ y'_k = \eta y_{k-1} + (1-\eta)y_k & k \geq 2. \end{cases} \quad (28)$$

where  $y'_k$  represents the processed data,  $y_k$  represents the original data, and  $\eta$  is a constant coefficient.

Then, we normalize the battery data to  $(-1, 1)$  to reduce the network calculation load and improve the training speed.

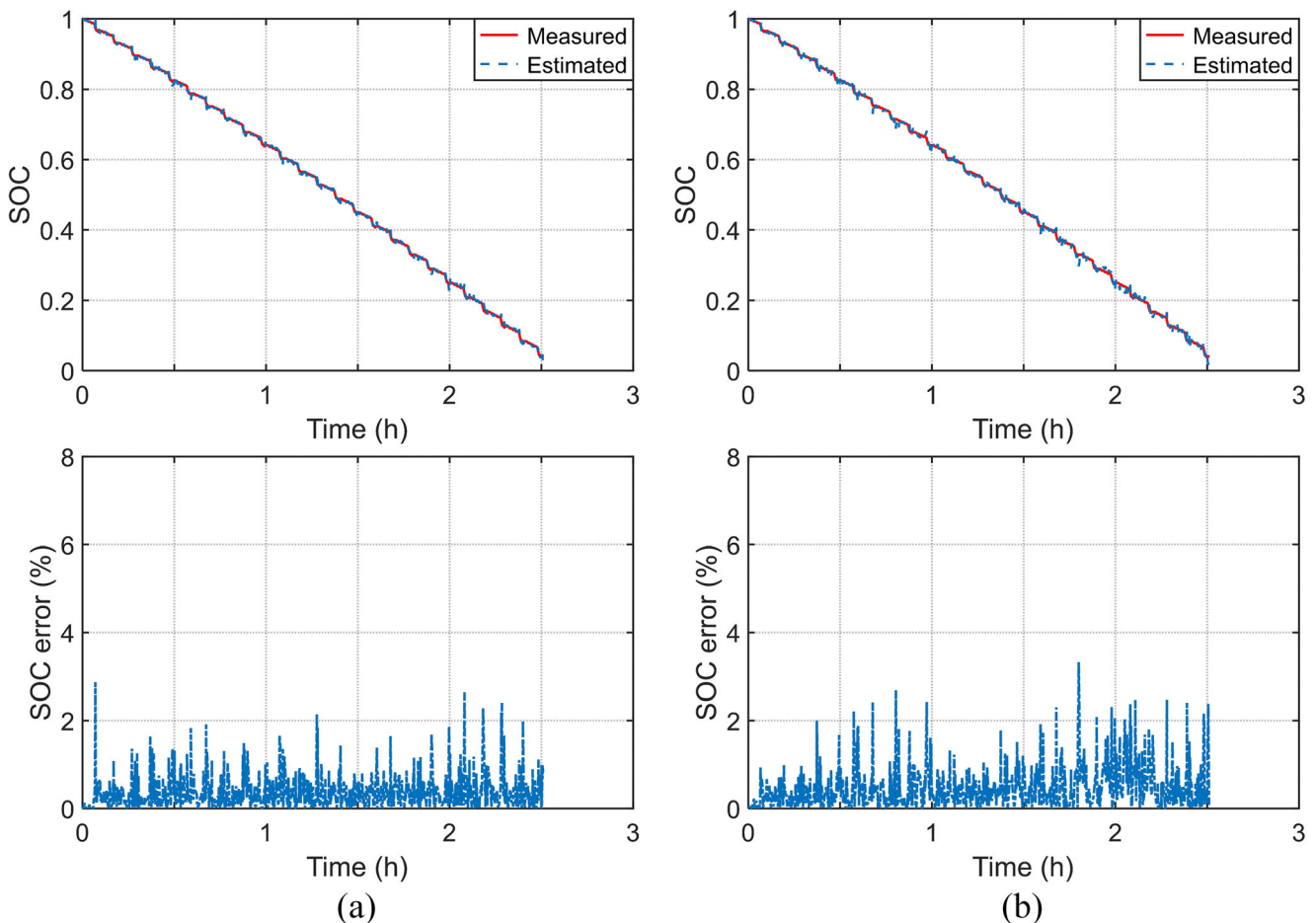
## Simulation

Take the mean squared error (MSE) as the cost function, and adopt the root mean squared error (RMSE), the mean absolute error (MAE), as well as the coefficient of determination ( $R^2$ ) to evaluate the estimation performance of the CG-RELM.

Randomly select 75% of all processed data as the training set, and the rest as the test set.

- (1) Set the number of hidden layer neurons ( $L$ ) to 300, and then, respectively, apply the CG-RELM and BP neural network (BP-NN) to estimate SOC. The simulation results are shown in Fig. 4 and Table 2.

It can be seen that the CG-RELM has faster convergence speed and better estimation performance when compared with the BP-NN.



**Fig. 6** The estimated SOC and errors of the CG-RELM with different data sets. **a** The estimated SOC and errors without noises **b** The estimated SOC and errors with noises



**Table 4** The performance of the CG-RELM with different data sets.

Data sets	Indexes	RMSE	MAE	R <sup>2</sup>
Without noises	Training set	0.0049	0.0035	0.9997
	Test set	0.0053	0.0038	0.9996
With noises	Training set	0.0061	0.0043	0.9995
	Test set	0.0068	0.0048	0.9994

(2) Respectively apply the CG-RELM with different numbers of hidden layer neurons ( $L = 200$  and  $L = 600$ ) to estimate SOC. The simulation results are shown in Fig. 5 and Table 3.

It can be seen from the results that the CG-RELM has better estimation performance when  $L$  is larger.

(3) In order to test the robustness of the CG-RELM, we add appropriate amount of noise to the original data. The noise variance corresponding to the voltage data is 0.01 and the current data 0.03. Respectively apply the CG-RELM ( $L = 500$ ) with different data sets to estimate SOC. The simulation results are shown in Fig. 6 and Table 4.

It can be seen from Fig. 6 and Table 4 that although the estimation error under the noise-added data has increased, the estimated SOC's are still within a satisfactory range.

Figures 4, 5, and 6 and Tables 2, 3, and 4 indicate that the CG-RELM has high accuracy and strong robustness.

## Conclusion

A conjugate gradient optimized regularized extreme learning machine (CG-RELM) is investigated to estimate battery SOC. In this hybrid algorithm, the conjugate gradient (CG) algorithm is used to calculate the output weights of the regularized extreme learning machine (RELM), which is a special feedforward neural network (SLFN) with fixed input weights and hidden layer biases. The weights adjustment directions at different points calculated by the CG algorithm are conjugate orthogonal to each other, avoiding the calculation load caused by the inverse matrix and guaranteeing the high convergence speed of the CG-RELM.

The simulation results verify that the CG-RELM can effectively estimate the battery SOC. (i) The convergence speed of the CG-RELM is faster when compared with the BP neural network. (ii) Increasing the number of hidden layer neurons can improve estimation precision. (iii) The algorithm shows high robustness when applied to the noise-added data set. This

investigated method can also be applied to block-oriented systems with NN non-linear parts [45–47].

**Funding** This work was supported by the National Natural Science Foundation of China under Grant Nos. 61873138.

## References

- Zubi G, Dufo-Lopez R, Carvalho M, Pasaoglu G (2018) The lithium-ion battery: state of the art and future perspectives. *Renewable and Sustainable Energy Reviews* 89:292–308
- Duan B, Li Z, Gu P, Zhou Z, Zhang C (2018) Evaluation of battery inconsistency based on information entropy. *Journal of Energy Storage* 16:160–166
- Lu L, Han X, Li J, Hua J, Ouyang M (2013) A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of Power Sources* 226:272–288
- Liu K, Li K, Peng Q, Zhang C (2019) A brief review on key technologies in the battery management system of electric vehicles. *Frontiers of Mechanical Engineering* 14:47–64
- Hong J, Wang Z, Chen W, Wang L, Qu C (2020) Online joint-prediction of multi-forward-step battery SOC using LSTM neural networks and multiple linear regression for real-world electric vehicles. *Journal of Energy Storage* 30: Art. no. 101459
- Lipu M, Hannan M, Hussain A, Hoque M, Ker P, Saad M, Ayob A (2018) A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: challenges and recommendations. *Journal of Cleaner Production* 205:115–133
- Li Z, Huang J, Liaw B, Zhang J (2017) On state-of-charge determination for lithium-ion batteries. *Journal of Power Sources* 348: 281–301
- Xiong R, Cao J, Yu Q, He H, Sun F (2018) Critical review on the battery state of charge estimation methods for electric vehicles. *IEEE Access* 6:1832–1843
- He H, Xiong R, Fan J (2011) Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. *Energies* 4(4):582–598
- Duan B, Zhang Q, Geng F, Zhang C (2020) Remaining useful life prediction of lithium-ion battery based on extended Kalman particle filter. *International Journal of Energy Research* 44(3):1724–1734
- Plett G (2004) Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 1. Background. *Journal of Power Sources* 134(2):252–261
- Plett G (2004) Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: part 2. Modeling and identification. *Journal of Power Sources* 134(2):262–276
- Plett G (2004) Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: part 3. State and parameter estimation. *Journal of Power Sources* 134(2):277–292
- Sun F, Hu X, Zou Y, Li S (2011) Adaptive unscented Kalman filtering for state of charge estimation of a lithium-ion battery for electric vehicles. *Energy* 36(5):3531–3540
- Li Y, Wang C, Gong J (2017) A multi-model probability SOC fusion estimation approach using an improved adaptive unscented Kalman filter technique. *Energy* 141:1402–1415
- Zeng Z, Tian J, Li D, Tian Y (2018) An online state of charge estimation algorithm for lithium-ion batteries using an improved adaptive cubature Kalman filter. *Energies* 11(1): Art. no. 59
- X. Cui, Z. Jing, M. Luo, Y. Guo, and H. Qiao (2018) A new method for state of charge estimation of lithium-ion batteries using square root cubature Kalman filter, *Energies*, vol. 11, no. 1, Art. no. 209.

18. Li W, Yang Y, Wang D, Yin S (2020) The multi-innovation extended Kalman filter algorithm for battery SOC estimation. *Ionics* 26(12):6145–6156
19. Chen J, Zhang Y, Zhu Q, Liu Y (2019) Aitken based modified Kalman filtering stochastic gradient algorithm for dual-rate nonlinear models. *Journal of Franklin Institute* 356(8):4732–4746
20. Feng L, Ding J, Han Y (2020) Improved sliding mode based EKF for the SOC estimation of lithium-ion batteries. *Ionics* 26(3):2875–2288
21. Muhammad S, Rafique M, Li S, Shao Z, Wang Q, Guan N (2017) A robust algorithm for state-of-charge estimation with gain optimization. *IEEE Transactions on Industrial Informatics* 13(6):2983–2994
22. Karmacharya IM, Gokaraju R (2018) Fault location in ungrounded photovoltaic system using wavelets and ANN. *IEEE Transactions on Power Delivery* 33(2):549–559
23. Lan S, Chen M, Chen D (2019) A novel HVDC double-terminal non-synchronous fault location method based on convolutional neural network. *IEEE Transactions on Power Delivery* 34(3):848–857
24. Bagheri A, Gu IYH, Bollen MHJ, Balouji E (2018) A robust transform-domain deep convolutional network for voltage dip classification. *IEEE Transactions on Power Delivery* 33(6):2794–2802
25. Peng X, Yang F, Wang G, Wu Y, Li L, Li Z, Bhatti AA, Zhou C, Hepburn DM, Reid AJ, Judd MD, Siew WH (2019) A convolutional neural network-based deep learning methodology for recognition of partial discharge patterns from high-voltage cables. *IEEE Transactions on Power Delivery* 34(4):1460–1469
26. Chemali E, Kollmeyer P, Preindl M, Emadi A (2018) State-of-charge estimation of Li-ion batteries using deep neural networks: a machine learning approach. *Journal of Power Sources* 400:242–255
27. Jiao M, Wang D, Qiu J (2020) A GRU-RNN based momentum optimized algorithm for SOC estimation. *Journal of Power Sources* 459: Art. no. 228051.
28. Nakama T (2009) Theoretical analysis of batch and on-line training for gradient descent learning in neural networks. *Neurocomputing* 34(1-3):151–159
29. Gan M, Guan Y, Chen G, Chen C (2020) Recursive variable projection algorithm for a class of separable nonlinear models. *IEEE Transactions on Neural Network and Learning Systems* (in press) Doi:<https://doi.org/10.1109/TNNLS.2020.3026482>
30. Huang G, Zhu Q, Siew C (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. *IEEE International Joint Conference on Neural Networks* 1-4:985–990
31. Huang G, Zhu Q, Siew C (2006) Extreme learning machine: theory and applications. *Neurocomputing* 70:489–501
32. Huang G, Li M, Chen L, Siew C (2008) Incremental extreme learning machine with fully complex hidden nodes. *Neurocomputing* 71(4-6):576–583
33. Luo X, Chang X, Ban X (2016) Regression and classification using extreme learning machine based on L-1-norm and L-2-norm. *Neurocomputing* 174: 179–186
34. Huang G (2015) What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. *Cognitive Computation* 7(3):263–278
35. Cosmo D, Salles E (2019) Multiple sequential regularized extreme learning machines for single image super resolution. *IEEE Signal Processing Letters* 26(3):440–444
36. Gumaei A, Hassan M, Hassan M, Alelaiwi A, Fortino F (2019) A hybrid feature extraction method with regularized extreme learning machine for brain tumor classification. *IEEE Access* 7:36266–36273
37. Weng F, Hou M, Zhang T, Yang Y, Wang Z, Sun H, Zhu H, Luo J (2018) Application of regularized extreme learning machine based on BIC criterion and genetic algorithm in iron ore price forecasting. 3rd International Conference on Modelling, Simulation and Applied Mathematics (MSAM), Shanghai, China, Jul 22-23, pp. 212–217
38. Li S, You Z, Guo H, Luo X, Zhao Z (2016) Inverse-free extreme learning machine with optimal information updating. *IEEE Transactions on Cybernetics* 46(5):1229–1241
39. Hestenes MR, Steifel EL (1952) Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards* 49(6):409–436
40. Polyak BT (1969) The conjugate gradient method in extremal problems. *USSR Computational Mathematics and Mathematical Physics* 9(4):94–112
41. Fletcher R, Reeves CM (1964) Function minimization by conjugate gradients. *The Computer Journal* 7(2):149–154
42. Li M, Li D A modified conjugate-descent method and its global convergence. *Pacific Journal of Optimization* 8(2):247–259
43. Shi Z, Shen J (2007) Convergence of Liu-Storey conjugate gradient method. *European Journal of Operational Research* 182(2):552–560
44. Dai Y (1999) Yuan Y (1999) A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on Optimization* 10(1):177–182
45. Wang D, Zhang S, Gan M, Qiu J (2020) A novel EM identification method for Hammerstein systems with missing output data. *IEEE Transactions on Industrial Informatics* 16(4):2500–2508
46. Wang D, Li L, Ji Y, Yan Y (2018) Model recovery for Hammerstein systems using the auxiliary model based orthogonal matching pursuit method. *Applied Mathematical Modelling* 54: 537–550
47. Wang D, Yan Y, Liu Y, Ding J (2019) Model recovery for Hammerstein systems using the hierarchical orthogonal matching pursuit method. *Journal of Computational and Applied Mathematics* 345:135–145

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.