Neural Networks 91 (2017) 22–33

Contents lists available at ScienceDirect

Neural Networks

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

Recursive least mean *p*-power Extreme Learning Machine

Jing Yang [a], Feng Ye [a], Hai-Jun Rong [b,*], Badong Chen [c]

[a] Institute of Control Engineering, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China
[b] State Key Laboratory for Strength and Vibration of Mechanical Structures, School of Aerospace, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China
[c] Institute of Artificial Intelligence and Robotics, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China

ARTICLE INFO

Article history:
Received 26 November 2016
Received in revised form 1 March 2017
Accepted 4 April 2017
Available online 12 April 2017

Keywords:
Recursive least mean *p*-power
Extreme learning machine
Online sequential learning
Non-Gaussian noises
Alpha-stable noises

ABSTRACT

As real industrial processes have measurement samples with noises of different statistical characteristics and obtain the sample one by one usually, on-line sequential learning algorithms which can achieve better learning performance for systems with noises of various statistics are necessary. This paper proposes a new online Extreme Learning Machine (ELM, of Huang et al.) algorithm, namely recursive least mean *p*-power ELM (RLMP-ELM). In RLMP-ELM, a novel error criterion for cost function, namely the least mean *p*-power (LMP) error criterion, provides a mechanism to update the output weights sequentially. The LMP error criterion aims to minimize the mean *p*-power of the error that is the generalization of the mean square error criterion used in the ELM. The proposed on-line learning algorithm is able to provide on-line predictions of variables with noises of different statistics and obtains better performance than ELM and online sequential ELM (OS-ELM) while the non-Gaussian noises impact the processes. Simulations are reported to demonstrate the performance and effectiveness of the proposed methods.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

A demand to build the predictive models with on-line variables is increasing in industry, economic sphere and other various fields (Chang & Fan, 2008; Chen, Zhao, Zhu, & Príncipe, 2012, 2013; Golestaneh, Pinson, & Gooi, 2016; Li, Jia, & Li, 2014), for instance, forecasting of renewable energy generation (Golestaneh et al., 2016), stock forecast (Chang & Fan, 2008), and weather forecast (Li et al., 2014). In these practical applications, the datum samples are often arriving in the order of time and contaminated by the large stochastic noises with different statistic characteristics, such as uniform, Gaussian, impulsive, or mixed distribution. Support vector machines (SVMs) (Sapankevych & Sankar, 2009; Trafalis & Ince, 2000), neural networks (Shi & Han, 2009; Wang & Han, 2014), and other machine learning methods (Qin, Nishii, & Yang, 2012; Sun, Zhang, & Yu, 2006) have been applied for prediction. However, most of the methods could obtain the best performance with the assumption of Gaussian noises or without noise. For both researchers and enterprise groups, on-line sequential learning algorithms, which are highly efficient and better learning performance for systems with various statistics, are keenly sought.

Over the past several decades, single layer feedforward networks (SLFNs) have been intensively studied as the basis for solving this problem (Ferrari & Stengel, 2005; Hou & Han, 2010; Meir & Maiorov, 2000). There have been a lot of learning algorithms for training SLFNs, including back-propagation (BP) algorithm and its various improved algorithms and so on (LeCun, Bottou, Orr, & Müller, 1998; MacKay, 1992). However, in these learning algorithms, all the parameters of SLFNs need to be tuned, which results in a slow learning speed and much training time when the number of training data is large. Recently, a new fast neural learning algorithm referred to as Extreme Learning Machine (ELM) has been developed for a SLFN with hidden neuron weights randomly initialized, which possesses universal approximation capability (Huang, Chen, & Siew, 2006; Huang, Zhu, & Siew, 2006). Compared with full parameter determination algorithms such as BP algorithm, the hidden nodes' random parameter initialization procedure with an analytical weight solution is computationally simple (Huang, Zhu et al., 2006).

ELM and most of its improved algorithms require all the training data available before training, that is, they are batch learning algorithms. For our on-line prediction problem, learning has to be an ongoing process since the complete set of data is usually not available at once. When some new data arrive, batch learning has to repeat the training with the past data as well as the new data, so it takes a lot of time. To handle this problem, online sequential ELM (OS-ELM) (Liang, Huang, Saratchandran, & Sundararajan, 2006)

* Corresponding author.
E-mail address: hjrong@xjtu.edu.cn (H.-J. Rong).

http://dx.doi.org/10.1016/j.neunet.2017.04.001
0893-6080/© 2017 Elsevier Ltd. All rights reserved.

and its different improvements have been successfully applied in some applications (Deng, Zheng, & Wang, 2014; Lim, Lee, & Pang, 2013; Matias, Gabriel, Souza, Araújo, & Pereira, 2013; Rong, Huang, Sundararajan, & Saratchandran, 2009; Soares & Araújo, 2016; Yang, Chen, Rong, & Chen, 2016; Yang, Shi, & Rong, 2016; Ye, Squartini, & Piazza, 2013).

Random noises widely exist in many practice systems, especially in sensor systems. Some variants of the ELM have been developed for noisy data. In Man, Lee, Wang, Cao, and Khoo (2012); Man, Lee, Wang, Cao, and Miao (2011), with the assumption of uniform noise, a finite impulse response ELM (FIR-ELM) and a discrete Fourier transform ELM (DFT-ELM) are proposed to improve the performance of ELM on the noisy input data. The input weights of FIR-ELM are assigned based on the FIR filter and the input weights of DFT-ELM are designed based on DFT technique, respectively. For solving the outlier robustness problem, ELM based on iteratively reweighted least squares (IRWLS-ELM), ELM based on the multivariate least rimmed squares (MLTS-ELM) and ELM based on the one-step reweighted MLTS (RMLTS-ELM) are proposed in Horata, Chiewchanwattana, and Sunat (2013) by iteratively reweighted training data and the multivariate least-trimmed squares estimator (MLTS). These above methods only consider the Gaussian noises or outlier noises without non-Gaussian noises in the whole process.

Mean squared error (MSE) criterion is exclusively adopted in these above ELM methods when constructing their cost functions. The MSE criterion makes sense in the linear signal processing with Gaussian assumption, because it only takes into account the second-order statistics. However, the MSE criterion may perform poorly in the data under nonlinear and non-Gaussian situations, as it captures only the second-order statistics in the samples. In many real-world circumstances, the data encountered are more impulsive in nature than that predicted by a Gaussian distribution, even impulsive and Gaussian distribution mixed, such as the energy spectrums of brain magnetic resonance (MR) images (Liao & Chung, 2010), multiple access interference (MAI) in communication systems (broadband power-line communications Beaulieu & Niranjayan, 2007; Zimmermann & Dostert, 2002, wireless sensor networks Lee & Tepedelenlioglu, 2014; Wen, 2013), noise of underwater acoustics (Bouvet & Schwartz, 1989; Nikias & Shao, 1995), audio processing (Kismode, 2000; Zhong, Premkumar, & Madhukumar, 2013), real time traffic prediction (Mao, 2005), low frequency atmospheric propagation (Chrissan, 1998), and other scenarios with man-made noise. These impulsive distribution problems, also known as the non-Gaussian heavy-tailed distribution problems, cannot be satisfactorily solved by the MSE criterion. On the other hand, in many real industrial production process, the measurement noise statistical characteristics of the instrument is the non-Gaussian light-tailed distribution, of which bounded uniform distribution is a particular case. At this time, the MSE criterion is also difficult to obtain the best performance.

In the field of adaptive filtering, in order to take into account lower-order or higher-order statistics, the least mean p-power (LMP) criterion has been studied. Typical examples include the least mean fourth (LMF) (Walach & Widrow, 1984) and LMP criterion (Pei & Tseng, 1994a), least mean mixed-norm (LMMN) criterion (Chambers, Tanrikulu, & Constantinides, 1994) and many others (Chen, Liu, Zhao, & Principe, 2017; Chen, Wang, Zhao, & Zheng, 2015; Chen, Xing, Liang, Zheng, & Príncipe, 2014; Chen et al., 2015; Chen, Xing, Zhao, Zheng, & Príncipe, 2016; Ma, Qu, Zhao, Chen, & Gui, 2015; Xiao, Tadokoro, & Shida, 1999). The LMP criterion uses the mean p-power error as the cost function ($\phi(e) = |e|^p, p \in \mathbb{N}$). It is computationally simple, and has been proven successful in numerous applications (Ma et al., 2015; Pei & Tseng, 1994a; Xiao et al., 1999). When used as an error criterion in adaptive filtering, the LMP may produce a better solution compared with the MSE if the performance function has different optimum solutions for various p. The steepest descent algorithm based on LMP error criterion with $p > 2$ (especially when $p = 4$) may have better convergence performance (i.e. achieve either faster convergence speed or lower misadjustment) while the noises are uniform distribution. Furthermore, the adaptive filter based on LMP error criterion with $p < 2$ is robust to impulsive noises (Chen et al., 2015; Ma et al., 2015; Pei & Tseng, 1994a).

In order to improve the robustness and accuracy of ELM algorithm that produces a poor and unreliable solution for on-line prediction problems when the output data are stained with various disturbances, we develop in this work a sequential and recursive extreme learning machine with a cost function formulated by the least mean p-power (LMP) error criterion where the L_p norm minimization of the error is considered. For simplicity, it is named as the recursive least mean p-power ELM (RLMP-ELM). Simulation results show that this proposed method with different p values has better and more stable solution compared with the existing ELM and OS-ELM learning algorithm.

The remainder of this paper is as follows. We provide a brief review of the ELM and LMP error criterion in Section 2. In Section 3, the proposed RLMP-ELM algorithm is described. The performance of this proposed algorithm is subsequently verified on different artificial datasets and real-world datasets in Section 4. Section 5 summarizes the conclusions from this study.

2. Preliminary

2.1. Extreme learning machine

Consider N arbitrary distinct samples $(\mathbf{x}_k, t_k)$, where $\mathbf{x}_k \in \mathbf{R}^n$ is the kth input vector and $t_k \in \mathbf{R}$ is the associated desired value. The output of an ELM with $\tilde{N}$ hidden nodes equals as

$$f(\mathbf{x}_k) = \sum_{i=1}^{\tilde{N}} \beta_i G(\mathbf{x}_k; \mathbf{c}_i, a_i) = \boldsymbol{\beta}^T \boldsymbol{G}_k, \quad k = 1, \ldots, N \tag{1}$$

where $\mathbf{c}_i$ and a_i are the learning parameters of hidden nodes, $\boldsymbol{\beta} \in \mathbf{R}^{\tilde{N}}$ and $\boldsymbol{G}_k \in \mathbf{R}^{\tilde{N}}$ are the output weight vector and the hidden nodes' output vector with respect to the input $\mathbf{x}_k$. In this work, the Gaussian activation function $G(\mathbf{x}_k; \mathbf{c}_i, a_i) = g(a_i\|\mathbf{x}_k - \mathbf{c}_i\|)$ is adopted to compute the output of hidden nodes. In ELM, the parameters of hidden nodes $\mathbf{c}_i$ and a_i are randomly set and are not subject to any optimization.

The output weight vector $\boldsymbol{\beta}_k$ is trained using the least mean square (LMS) algorithm based on the minimization of the following mean square error (MSE) cost function:

$$J_{\text{MSE}} = \frac{1}{N}\sum_{k=1}^{N} e_k^2 = \frac{1}{N}\|\mathbf{H}\boldsymbol{\beta} - \boldsymbol{T}\| = E(e_k^2) \tag{2}$$

where E denotes the expectation operator, $e_k = t_k - \boldsymbol{\beta}^T \boldsymbol{G}_k$ is the estimation error. H denotes the hidden layer output matrix, where $h_{ki} \in \mathbf{H}(k = 1, \ldots, N; i = 1, \ldots, \tilde{N})$ is the activation value of the ith hidden neuron for the kth input vector $h_{ki} = g(a_i\|\mathbf{x}_k - \mathbf{c}_i\|)$. $\boldsymbol{T}(= [t_1, \ldots, t_k, \ldots, t_N]^T)$ is the desired output vector. A pseudoinverse operation yields the unique L_2 solution of (2), that is $\boldsymbol{\beta} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\boldsymbol{T}$.

However, the MSE criterion may perform poorly in many situations, especially in nonlinear and non-Gaussian situations, as it captures only the second-order statistics in the data. In order to take into account higher-order (or lower-order) statistics and to improve the robust performance in realistic scenarios, an alternative optimality criterion beyond the second-order statistics has been applied in our study.

2.2. Least mean p-power

Let $e_k = t_k - f(\mathbf{x_k})$ be the estimation error. Then, the least mean p-power (LMP) cost is defined as ($p \in \mathbb{N}$),

$$J_{\text{LMP}} = \min E(|e_k|^p) \tag{3}$$

which includes the MSE criterion as a special case. When $p = 2$, this criterion reduces to the MSE criterion (2). The LMP criterion is computationally simple, and has been proven successful in various applications. Many literatures (Chen et al., 2015; Chen, Zhu, Hu, & Principe, 2013; Ma et al., 2015; Pei & Tseng, 1994a, 1994b; Wen, 2013; Xiao & Shida, 1999; Xiao et al., 1999) have pointed out that the LMP has some useful properties such that it may produce a better solution if the performance function has different optimum solutions for various p, instead of the MSE; while the datum is non-Gaussian light-tailed distribution, steepest descent algorithm based on LMP error criterion with $p > 2$ (especially when $p = 4$) may have better convergence performance (i.e. achieve either faster convergence speed or lower misadjustment); the learning algorithm based on LMP error criterion with $p < 2$ (e.g. when $p = 1$) is robust to non-Gaussian heavy-tailed distribution noises.

3. Recursive least mean p-power extreme learning machine

An empirical least mean p-power related sequential extreme learning machine (RLMP-ELM) is developed in this section. The RLMP-ELM is based on the primitive ELM algorithm which is essentially a randomly parameterized SLFN construction. The ELM learning operation is replaced by a recursive least mean p-power sequential updating procedure in the RLMP-ELM. In this section, we will derive the algorithm to update the weight vector of the ELM under the LMP error criterion (3). In the following parts, we will present the detailed process of the RLMP-ELM algorithm.

3.1. RLMP-ELM algorithm

According to the depiction in 2.1, the output layer of an ELM can be seen as a general linear system $\boldsymbol{\beta}^T\boldsymbol{G} = t$, where $\boldsymbol{\beta} \in \mathbf{R}^{\tilde{N}}$, $\boldsymbol{G} \in \mathbf{R}^{\tilde{N}}$ and $t \in \mathbf{R}$. And now, we need to estimate the value of $\boldsymbol{\beta}$. For this general linear system, the RLMP algorithm is the extensive of the recursive least square (RLS) algorithm with cost function (2) (Bhotto & Antoniou, 2011; Chan & Zou, 2004; Zha, 2006). The cost function of LMP algorithm is,

$$J_{LMP} = \frac{1}{N}\sum_{k=1}^{N}|e_k|^p \tag{4}$$

where e_k is the error in kth sample time and $e_k = t_k - \boldsymbol{\beta}_N^T\boldsymbol{G}_k$. In theory, it has been proved by some results of convex function in the literature (Pei & Tseng, 1994b) that the every minimum of performance function J_{LMP} is a global minimum while $p \geq 1$. The optimal solution $\boldsymbol{\beta}_N$ for minimizing J_{LMP} can be obtained by differentiating Eq. (4) with respect to $\boldsymbol{\beta}_N$ and setting the derivatives to zero. The derivatives are

$$\begin{aligned}\frac{\partial J_{LMP}}{\partial \boldsymbol{\beta}_N} &= \frac{1}{N}\sum_{k=1}^{N}\frac{\partial|e_k|^p}{\partial \boldsymbol{\beta}_N} \\ &= \frac{1}{N}\sum_{k=1}^{N}\frac{\partial|e_k|^p}{\partial e_k}\cdot\frac{\partial e_k}{\partial \boldsymbol{\beta}_N}.\end{aligned} \tag{5}$$

Also, because

$$|e_k|^p = \begin{cases} e_k^p & p : \text{even} \\ \text{sgn}(e_k)e_k^p & p : \text{odd} \end{cases} \tag{6}$$

the following expression is obtained:

$$\begin{aligned}\frac{\partial|e_k|^p}{\partial e_k} &= \begin{cases} pe_k^{p-1} & p : \text{even} \\ p\text{sgn}(e_k)e_k^{p-1} & p : \text{odd} \end{cases} \\ &= p|e_k|^{p-2}e_k\end{aligned} \tag{7}$$

where $\text{sgn}(e_k) = e_k/|e_k|$. Thus, Eq. (5) can be written as follows:

$$\frac{\partial J_{LMP}}{\partial \boldsymbol{\beta}_N} = \frac{1}{N}\sum_{k=1}^{N}p|e_k|^{p-2}e_k\frac{\partial e_k}{\partial \boldsymbol{\beta}_N}. \tag{8}$$

Substituting $e_k = t_k - \boldsymbol{\beta}_N^T\boldsymbol{G}_k$ into Eq. (8) yields

$$\frac{\partial J_{LMP}}{\partial \boldsymbol{\beta}_N} = \frac{1}{N}\sum_{k=1}^{N}p|e_k|^{p-2}(t_k - \boldsymbol{\beta}_N^T\boldsymbol{G}_k)\boldsymbol{G}_k. \tag{9}$$

Setting $\frac{\partial J_{LMP}}{\partial \boldsymbol{\beta}_N} = 0$, Eq. (9) can be further written as follows:

$$\sum_{k=1}^{N}|e_k|^{p-2}\boldsymbol{G}_k\boldsymbol{G}_k^T\boldsymbol{\beta}_N = \sum_{k=1}^{N}|e_k|^{p-2}t_k\boldsymbol{G}_k. \tag{10}$$

Let

$$\mathbf{R}_N = \sum_{k=1}^{N}|e_k|^{p-2}\boldsymbol{G}_k\boldsymbol{G}_k^T \tag{11}$$

and

$$\boldsymbol{P}_N = \sum_{k=1}^{N}|e_k|^{p-2}t_k\boldsymbol{G}_k. \tag{12}$$

Here, we set $\mathbf{G}_N = [\boldsymbol{G}_1, \ldots, \boldsymbol{G}_N]$, then $\mathbf{R}_N$ and $\boldsymbol{P}_N$ are called the p-Power correlation matrix of $\mathbf{G}_N$ and the p-Power cross-correlation vector of $\mathbf{G}_N$ and $\boldsymbol{T}$, respectively. They serve similar purpose as the conventional correlation matrix of $\mathbf{G}_N$ and the cross-correlation vector of $\mathbf{G}_N$ and $\boldsymbol{T}$.

According to Eq. (10), the following relation can be obtained:

$$\mathbf{R}_N\boldsymbol{\beta}_N = \boldsymbol{P}_N. \tag{13}$$

The optimal solution $\boldsymbol{\beta}_N$ is

$$\boldsymbol{\beta}_N = \mathbf{R}_N^{-1}\boldsymbol{P}_N. \tag{14}$$

Eqs. (11) and (12) can be further written as

$$\begin{aligned}\mathbf{R}_N &= \sum_{k=1}^{N-1}|e_k|^{p-2}\boldsymbol{G}_k\boldsymbol{G}_k^T + |e_N|^{p-2}\boldsymbol{G}_N\boldsymbol{G}_N^T \\ &= \mathbf{R}_{N-1} + |e_N|^{p-2}\boldsymbol{G}_N\boldsymbol{G}_N^T\end{aligned} \tag{15}$$

$$\begin{aligned}\boldsymbol{P}_N &= \sum_{k=1}^{N-1}|e_k|^{p-2}t_k\boldsymbol{G}_k + |e_N|^{p-2}t_N\boldsymbol{G}_N \\ &= \boldsymbol{P}_{N-1} + |e_N|^{p-2}t_N\boldsymbol{G}_N.\end{aligned} \tag{16}$$

Substituting Eq. (16) into Eq. (14), we can get

$$\boldsymbol{\beta}_N = \mathbf{R}_N^{-1}(\boldsymbol{P}_{N-1} + |e_N|^{p-2}t_N\boldsymbol{G}_N). \tag{17}$$

According to Eq. (14), there is

$$\boldsymbol{P}_{N-1} = \mathbf{R}_{N-1}\boldsymbol{\beta}_{N-1}. \tag{18}$$

From Eq. (15), the following can be obtained:

$$\mathbf{R}_{N-1} = \mathbf{R}_N - |e_N|^{p-2}\boldsymbol{G}_N\boldsymbol{G}_N^T. \tag{19}$$

Substituting Eq. (19) into Eq. (18), and then taking the result of $\boldsymbol{P}_{N-1}$ into Eq. (17), we can get

$$\begin{aligned}\boldsymbol{\beta}_N &= \mathbf{R}_N^{-1}[(\mathbf{R}_N - |e_N|^{p-2}\boldsymbol{G}_N\boldsymbol{G}_N^T)\boldsymbol{\beta}_{N-1} + |e_N|^{p-2}t_N\boldsymbol{G}_N] \\ &= \mathbf{R}_N^{-1}(\mathbf{R}_N\boldsymbol{\beta}_{N-1} - |e_N|^{p-2}\boldsymbol{G}_N\boldsymbol{G}_N^T\boldsymbol{\beta}_{N-1} + |e_N|^{p-2}t_N\boldsymbol{G}_N) \\ &= \boldsymbol{\beta}_{N-1} + |e_N|^{p-2}\mathbf{R}_N^{-1}\boldsymbol{G}_N(t_N - \boldsymbol{G}_N^T\boldsymbol{\beta}_{N-1}). \end{aligned} \tag{20}$$

The equation for updating $\boldsymbol{\beta}_N$ can be obtained as follows:

$$\boldsymbol{\beta}_N = \boldsymbol{\beta}_{N-1} + |e_N|^{p-2}\mathbf{R}_N^{-1}\boldsymbol{G}_N(t_N - \boldsymbol{\beta}_{N-1}^T\boldsymbol{G}_N). \tag{21}$$

Applying the matrix inversion lemma (Diniz, 2008),

$$(\mathbf{A} + \mu\boldsymbol{x}\boldsymbol{y}^T)^{-1} = \mathbf{A}^{-1}\left(\mathbf{I} - \frac{\mu\boldsymbol{x}\boldsymbol{y}^T\mathbf{A}^{-1}}{1 + \mu\boldsymbol{y}^T\mathbf{A}^{-1}\boldsymbol{x}}\right) \tag{22}$$

and letting $\mathbf{R}_{N-1} = \mathbf{A}, \boldsymbol{x} = \boldsymbol{y} = G_N, \mu = |e_N|^{p-2}$. According Eq. (15), we can obtain

$$\mathbf{R}_N^{-1} = \left(\mathbf{I} - \frac{|e_N|^{p-2}\mathbf{R}_{N-1}^{-1}\boldsymbol{G}_N}{1 + |e_N|^{p-2}\boldsymbol{G}_N^T\mathbf{R}_{N-1}^{-1}\boldsymbol{G}_N}\boldsymbol{G}_N^T\right)\mathbf{R}_{N-1}^{-1}. \tag{23}$$

In ELM, select the type of nodes (additive or RBF) and the corresponding activation function g and the hidden node number $\tilde{N}$. The data $\aleph = \{(\mathbf{x}_k, t_k)|\mathbf{x}_k \in \mathbf{R}^n, t_k \in \mathbf{R}, k = 1, \ldots, N_0\}$ arrives already and the new data follows sequentially. The initial value of $\boldsymbol{\beta}_0$ is set as zero. Here, we set

$$\mathbf{H}_0 = \begin{bmatrix}\boldsymbol{G}_1^T \\ \vdots \\ \boldsymbol{G}_{N_0}^T\end{bmatrix} = \begin{bmatrix} g(\mathbf{x}_1; \mathbf{c}_1, a_1) & \ldots & g(\mathbf{x}_1; \mathbf{c}_{\tilde{N}}, a_{\tilde{N}}) \\ & \vdots & \\ g(\mathbf{x}_{N_0}; \mathbf{c}_1, a_1) & \ldots & g(\mathbf{x}_{N_0}; \mathbf{c}_{\tilde{N}}, a_{\tilde{N}})\end{bmatrix}_{N_0\times\tilde{N}} \tag{24}$$

$$\boldsymbol{T}_0 = [t_1\ t_2\ \ldots\ t_{N_0}]^T \tag{25}$$

$$\mathbf{E}_0 = \begin{bmatrix} |t_1|^{\frac{p}{2}-1} & \ldots & 0 \\ & \vdots & \\ 0 & \ldots & |t_{N_0}|^{\frac{p}{2}-1}\end{bmatrix}_{N_0\times N_0}. \tag{26}$$

Letting $\mathbf{M}_0 = \mathbf{E}_0\mathbf{H}_0$, we get

$$\mathbf{R}_0 = \mathbf{M}_0^T\mathbf{M}_0 \tag{27}$$

$$\boldsymbol{P}_0 = \mathbf{H}_0^T\boldsymbol{T}_0 \tag{28}$$

$$\mathbf{R}_0^{-1} = (\mathbf{M}_0^T\mathbf{M}_0)^{-1} \tag{29}$$

$$\boldsymbol{\beta}_0 = (\mathbf{M}_0^T\mathbf{M}_0)^{-1}\mathbf{H}_0^T\boldsymbol{T}_0. \tag{30}$$

For the new arriving data $(\mathbf{x}_1, t_1)$, we can get the new hidden layer output $\boldsymbol{H}_1$ and update the parameters of output layer $\boldsymbol{\beta}_1$ according to Eqs. (21) and (23):

$$\boldsymbol{H}_1 = [g(\mathbf{x}_1;\ \mathbf{c}_1, a_1)\, g(\mathbf{x}_1; \mathbf{c}_2, a_2), \ldots, g(\mathbf{x}_1;\ \mathbf{c}_{\tilde{N}}, a_{\tilde{N}})] \tag{31}$$

$$\mathbf{R}_1^{-1} = \left(\mathbf{I} - \frac{|e_1|^{p-2}\mathbf{R}_0^{-1}\boldsymbol{H}_1^T}{1 + |e_1|^{p-2}\boldsymbol{H}_1\mathbf{R}_0^{-1}\boldsymbol{H}_1^T}\boldsymbol{H}_1\right)\mathbf{R}_0^{-1} \tag{32}$$

$$\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 + |e_1|^{p-2}\mathbf{R}_1^{-1}\boldsymbol{H}_1^T(t_1 - \boldsymbol{H}_1\boldsymbol{\beta}_0) \tag{33}$$

where $e_1 = t_1 - \boldsymbol{H}_1\boldsymbol{\beta}_0$. If there arrives any other new data, then Eqs. (31)–(33) can be repeated.

3.2. Universal approximation of RLMP-ELM algorithm

Consider again the description of ELM in Section 2.1, where there are a standard SLFN and N arbitrary distinct samples $(\mathbf{x}_k, t_k)$ in the algorithm. The SLFN with $\tilde{N}$ hidden nodes with activation function $g(x)$ can approximate these N samples with zero error means that $\sum_{k=1}^{N}|t_k - \boldsymbol{\beta}_N^T\boldsymbol{G}_k|^p = 0$, where $\boldsymbol{G}_k = [g(\mathbf{x}_k; \mathbf{c}_1, a_1), g(\mathbf{x}_k; \mathbf{c}_2, a_2), \ldots, g(\mathbf{x}_k; \mathbf{c}_{\tilde{N}}, a_{\tilde{N}})]^T$.

If the activation function g is infinitely differentiable, we can prove that the required number of hidden nodes $\tilde{N} \leq N$. Strictly speaking, we have

Theorem 1. *Given a standard SLFN with $\tilde{N}$ hidden nodes and activation function $g : R \to R$ which is infinitely differentiable in any interval, for N arbitrary distinct samples $(\mathbf{x}_k, t_k)$, where $\mathbf{x}_k \in \mathbf{R}^n$ and $t_k \in \mathbf{R}$, for any $\mathbf{c}_i$ and a_i randomly chosen from any intervals of $\mathbf{R}^n$ and $\mathbf{R}$, respectively, according to any continuous probability distribution, then with probability one, $\mathbf{R}_N$, the p-power correlation matrix of $\mathbf{G}_N$, is invertible and $\sum_{k=1}^{N}|t_k - \boldsymbol{\beta}_N^T\boldsymbol{G}_k|^p = 0$.*

Proof. From Eq. (23), it can be seen that $\mathbf{R}_0$ should be full rank so that it could be invertible, namely, $rank(\mathbf{R}_0) = \tilde{N}$. Eq. (11) can be rewritten as

$$\boldsymbol{R}_N = [\boldsymbol{G}_1 \ldots \boldsymbol{G}_N]\begin{bmatrix} |e_1|^{p-2} & \ldots & 0 \\ & \vdots & \\ 0 & \cdots & |e_N|^{p-2}\end{bmatrix}\begin{bmatrix}\boldsymbol{G}_1^T \\ \vdots \\ \boldsymbol{G}_N^T\end{bmatrix} \tag{34}$$

Let

$$\mathbf{G} = [\boldsymbol{G}_1 \ldots \boldsymbol{G}_N]\begin{bmatrix} |e_1|^{p/2-1} & \ldots & 0 \\ & \vdots & \\ 0 & \cdots & |e_N|^{p/2-1}\end{bmatrix}. \tag{35}$$

Eq. (34) can be transformed to $\mathbf{R}_N = \mathbf{G}\mathbf{G}^T$. If there are distinct N samples $\boldsymbol{G}_k$ and $\tilde{N} \leq N$, $rank([\boldsymbol{G}_1 \ldots \boldsymbol{G}_N]) = \tilde{N}$ and $rank(\mathbf{G}) = \tilde{N}$. According to the lemma in Zhang (2004), $rank(\mathbf{A}\mathbf{A}^T) = rank(\mathbf{A}^T\mathbf{A}) = rank(\mathbf{A})$, we can get $rank(\mathbf{R}_N) = \tilde{N}$.

In the theory of ELM, $\mathbf{H}$ is the hidden layer output matrix of the SLFN and $\mathbf{H} = [\boldsymbol{G}_1 \ldots \boldsymbol{G}_N]^{\mathbf{T}}$. According to Theorem 2.1 in Huang's literature (Huang, Zhu et al., 2006), since $\mathbf{c}_i$ and a_i are randomly chosen according to any continuous probability distribution, the column vectors of $\mathbf{H}$ can be made full-rank with probability one, namely $rank(\mathbf{H}) = \tilde{\mathbf{N}}$. Such activation functions include the sigmoidal functions as well as the radial basis, sine, cosine, exponential, and many other irregular functions.

Thus, $\mathbf{G}$ in Eq. (35) can also be made full-rank and according to the lemma of matrix rank in Zhang (2004), $\mathbf{R}_N$ is also full-rank. Following the derivation (4)–(14), $\sum_{k=1}^{N}|t_k - \boldsymbol{\beta}_N^T\boldsymbol{G}_k|^p = 0$ can be obtained obviously.

Similar to OS-ELM (Liang et al., 2006), the proposed RLMP-ELM algorithm consists of two phases, namely an initialization phase and a sequential learning phase. In the initialization phase, the value of $\boldsymbol{\beta}_0$ is estimated based on a small chunk of samples. Now, the RLMP algorithm for nonlinear system can be summarized as follows.

RLMP-ELM Algorithm:

Step 1 **Initialization Phase:** Initialize the learning using a small chunk of initial training data $\aleph_0 = \{(\mathbf{x}_k, t_k)\}_{k=1}^{N_0}$, $N_0 \geq \tilde{N}$.
- (a) Assign random input weights $\mathbf{c}_i$ and bias a_i (for additive hidden nodes) or center $\mathbf{c}_i$ and impact factor a_i (for RBF hidden nodes), $i = 1, \ldots, \tilde{N}$.
- (b) Calculate the initial hidden layer output matrix $\mathbf{H}_0$ according to Eq. (24) and $\boldsymbol{T}_0$ based on Eq. (25).
- (c) estimate the initial output weight $\boldsymbol{\beta}_0$ according to Eq. (30). Set the training step $k = 1$.

Step 2 **Sequential Learning Phase:**
- (a) Obtain the current training data $(\mathbf{x}_k, t_k)$.
- (b) Calculate the partial hidden layer output matrix $\boldsymbol{H}_k = [g(\mathbf{x}_k;\ \mathbf{c}_1, a_1)\, g(\mathbf{x}_k; \mathbf{c}_2, a_2), \ldots, g(\mathbf{x}_k;\ \mathbf{c}_{\tilde{N}}, a_{\tilde{N}})]$.

(c) Calculate the error term $e_k = t_k - \boldsymbol{H}_k\boldsymbol{\beta}_{k-1}$.
(d) Calculate the output weight $\boldsymbol{\beta}_k$

$$\begin{aligned}\mathbf{R}_k^{-1} &= \mathbf{R}_{k-1}^{-1} - \frac{|e_k|^{p-2}\mathbf{R}_{k-1}^{-1}\boldsymbol{H}_k^T}{1+|e_k|^{p-2}\boldsymbol{H}_k\mathbf{R}_{k-1}^{-1}\boldsymbol{H}_k^T}\boldsymbol{H}_k\mathbf{R}_{k-1}^{-1}\\ \boldsymbol{\beta}_k &= \boldsymbol{\beta}_{k-1} + |e_k|^{p-2}\mathbf{R}_k^{-1}\boldsymbol{H}_k^T e_k.\end{aligned} \tag{36}$$

(e) If there is any new training data, set $k = k+1$ and go to *step* 2. Otherwise, the algorithm is ended.

Remark 1. $\mathbf{R}_0$ and $\mathbf{G}_0$ in the initialization phase is expressed as follows:

$$\mathbf{R}_0 = [\boldsymbol{G}_1 \ \dots \ \boldsymbol{G}_{N_0}]\begin{bmatrix} |t_1|^{p-2} & \dots & 0 \\ & \vdots & \\ 0 & \cdots & |t_{N_0}|^{p-2}\end{bmatrix}\begin{bmatrix}\boldsymbol{G}_1^T \\ \vdots \\ \boldsymbol{G}_{N_0}^T\end{bmatrix}. \tag{37}$$

Let

$$\mathbf{G}_0 = [\boldsymbol{G}_1 \dots \boldsymbol{G}_{N_0}]\begin{bmatrix} |t_1|^{p/2-1} & \dots & 0 \\ & \vdots & \\ 0 & \cdots & |t_{N_0}|^{p/2-1}\end{bmatrix}. \tag{38}$$

According to Theorem 1, $\mathbf{R}_0$ is invertible while there are $\tilde{N}$ arbitrary distinct samples $(\mathbf{x}_k, t_k)$. If the first $\tilde{N}$ training data are not distinct, more training data are required and $\tilde{N} \le N_0$ is set. In most training cases, N_0 is equal to $\tilde{N}$ or close to $\tilde{N}$.

Remark 2. We can further compare the computation complexity between the proposed RLMP-ELM with the ELM and OS-ELM algorithms. For the $\tilde{N}$ hidden units and N-length training sequence, the total training complexity of the RLMP-ELM is of $O(N\tilde{N}^2)$. The same computation complexity can thus be observed comparing that of $O(N\tilde{N}^2)$ in the primitive ELM matrix inversion (Huang, Zhu et al., 2006) and of $O(N\tilde{N}^2)$ in the OS-ELM (Huang, Liang, Rong, Saratchandran, & Sundararajan, 2005; Liang et al., 2006). But the data is processed sequentially in the algorithm RLMP-ELM and OS-ELM, it costs more time in these two algorithms than it does in ELM algorithm.

Remark 3. Similar to OS-ELM, the output weight learning phase of the proposed RLMP-ELM can commence in a chunk-by-chunk learning mode. Here, we set

$$\mathbf{H}_{B_k} = \begin{bmatrix} g(\mathbf{x}_{N-B_k+1};\mathbf{c}_1,a_1) & \dots & g(\mathbf{x}_{N-B_k+1};\mathbf{c}_{\tilde{N}},a_{\tilde{N}}) \\ & \vdots & \\ g(\mathbf{x}_N;\mathbf{c}_1,a_1) & \dots & g(\mathbf{x}_N;\mathbf{c}_{\tilde{N}},a_{\tilde{N}})\end{bmatrix}_{B_k\times\tilde{N}} \tag{39}$$

$$\boldsymbol{T}_{B_k} = [t_{N-B_k+1}\ t_{N-B_k+2}\ \dots\ t_N]^T_{B_k\times 1} \tag{40}$$

$$\mathbf{E}_{B_k} = \begin{bmatrix} |e_{N-B_k+1}|^{\frac{p}{2}-1} & \dots & 0 \\ & \vdots & \\ 0 & \dots & |e_N|^{\frac{p}{2}-1}\end{bmatrix}_{B_k\times B_k}. \tag{41}$$

Let $\mathbf{M}_{B_k} = \mathbf{E}_{B_k}\mathbf{H}_{B_k}$. Moreover, according to Eq. (22), we get

$$\mathbf{R}_N^{-1} = \left(\mathbf{I} - \frac{\mathbf{R}_{N-B_k}^{-1}\mathbf{M}_{B_k}^T\mathbf{M}_{B_k}}{1+\mathbf{M}_{B_k}\mathbf{R}_{N-B_k}^{-1}\mathbf{M}_{B_k}^T}\right)\mathbf{R}_{N-B_k}^{-1} \tag{42}$$

where $\mathbf{R}_{N-B_k} = \mathbf{A}$, $\boldsymbol{x} = \boldsymbol{y} = \mathbf{M}_{B_k}^T$, $\mu = 1$. With the same derivation of Eq. (21), the update of the output weight is as follows:

$$\boldsymbol{\beta}_N = \boldsymbol{\beta}_{N-B_k} + \mathbf{R}_N^{-1}\mathbf{M}_{B_k}^T(\boldsymbol{T}_{B_k} - \mathbf{M}_{B_k}\boldsymbol{\beta}_{N-B_k}). \tag{43}$$

Also, the error terms equal as follows:

$$e_{N-B_k+l} = t_{N-B_k+l} - f(\mathbf{x}_{N-B_k+l}), \quad l = 1,\dots,B_k \tag{44}$$

where $f(\mathbf{x}_{N-B_k+l}) = \mathbf{H}_{B_k}^l\boldsymbol{\beta}_{N-B_k}$. $\mathbf{H}_{B_k}^l$ is the lth row of $\mathbf{H}_{B_k}$ and B_k is the block length. The chunk length may be of varying size, i.e., the number B_k of the inputs in the kth chunk does not need to be the same as B_{k-1}.

Remark 4. If there are N arbitrary distinct samples $(\mathbf{x}_k, \mathbf{t}_k)$, where $\mathbf{x}_k \in \mathbf{R}^n$ and $\mathbf{t}_k \in \mathbf{R}^m$, rather than $t_k \in \mathbf{R}$. The output of an ELM with $\tilde{N}$ hidden nodes equals as

$$\begin{aligned}\mathbf{f}_k(\mathbf{x}_k) &= \left[\sum_{i=1}^{\tilde{N}}\beta_{1i}G(\mathbf{x}_k;\mathbf{c}_i,a_i)\ \dots\ \sum_{i=1}^{\tilde{N}}\beta_{mi}G(\mathbf{x}_k;\mathbf{c}_i,a_i)\right]^T\\ &= [\boldsymbol{\beta}_1^T\boldsymbol{G}_k\ \dots\ \boldsymbol{\beta}_m^T\boldsymbol{G}_k]^T, \quad k = 1,\dots,N.\end{aligned} \tag{45}$$

where $\boldsymbol{\beta}_j \in R^{\tilde{N}}(j = 1,\dots,m)$. Let $\boldsymbol{\beta}_M = [\boldsymbol{\beta}_1\ \dots\ \boldsymbol{\beta}_m]$. The cost function of LMP algorithm is

$$J_{LMP} = \frac{1}{N}\sum_{k=1}^{N}\left(\sum_{j=1}^{m}|e_{jk}|^p\right). \tag{46}$$

Let $\mathbf{e_k} = [e_{1k}\ \dots\ e_{mk}]^T$ and $\mathbf{e_k} = \mathbf{t_k} - \mathbf{f}(\mathbf{x_k}) = \mathbf{t_k} - \boldsymbol{\beta}_{MN}^T\boldsymbol{G}_k$. The optimal solution $\boldsymbol{\beta}_{MN}$ for minimizing J_{LMP} can be obtained by differentiating Eq. (46) with respect to $\boldsymbol{\beta}_{MN}$ and setting the derivatives to zero. The derivatives are

$$\begin{aligned}\frac{\partial J_{LMP}}{\partial\boldsymbol{\beta}_{MN}} &= \frac{1}{N}\sum_{k=1}^{N}\frac{\sum_{j=1}^{m}|e_{jk}|^p}{\partial\boldsymbol{\beta}_{MN}}\\ &= \frac{1}{N}\left[\sum_{k=1}^{N}\frac{\partial|e_{1k}|^p}{\partial e_{1k}}\cdot\frac{\partial e_{1k}}{\partial\boldsymbol{\beta}_{1N}}\ \dots\ \sum_{k=1}^{N}\frac{\partial|e_{mk}|^p}{\partial e_{mk}}\cdot\frac{\partial e_{mk}}{\partial\boldsymbol{\beta}_{mN}}\right]\\ &= \frac{1}{N}\left[\sum_{k=1}^{N}p|e_{1k}|^{p-2}(t_{1k}-\boldsymbol{\beta}_{1N}^T\boldsymbol{G}_k)\boldsymbol{G}_k\ \dots\right.\\ &\quad\left.\times\sum_{k=1}^{N}p|e_{mk}|^{p-2}(t_{mk}-\boldsymbol{\beta}_{mN}^T\boldsymbol{G}_k)\boldsymbol{G}_k\right].\end{aligned} \tag{47}$$

Setting $\frac{\partial J_{LMP}}{\partial\boldsymbol{\beta}_{MN}} = 0$ and according to Eqs. (21) and (23), we can get the recursive formula for every $\boldsymbol{\beta}_{jN}(j = 1,\dots,m)$, respectively.

4. Performance evaluation

In this section, the performance of the proposed RLMP-ELM learning algorithm is compared with ELM and OS-ELM on quite a few benchmark real problems in the regression and time series prediction areas. To confirm the validity of the proposed RLMP-ELM with different p values, four different types of training samples have been used, respectively. Furthermore, we utilize the training samples with the noises of several different distributions for illustrating that the better performance could be achieved by choosing p value according to the features of the noises distribution.

Besides Gaussian noises, some non-Gaussian noises are considered. The symmetric alpha-stable ($S\alpha S$) distribution can model impulsive type of noises with heavy-tailed distributions (Nikias & Shao, 1995). It has been shown that the impulsive characteristics of many physical noise sources can be greatly captured by the $S\alpha S$ model (Beaulieu & Niranjayan, 2007; Kismode, 2000; Lee & Tepedelenlioglu, 2014; Liao & Chung, 2010; Wen, 2013; Zhong et al.,

2013). Generally, a $S\alpha S$ random distribution can be described conveniently by its characteristic function (Nikias & Shao, 1995; Shao & Nikias, 1993)

$$\phi(t) = \exp(j\mu t - \gamma|t|^{\alpha}) \tag{48}$$

where $\alpha \in (0, 2]$ is the characteristic exponent and completely determines the shape of the distribution, i.e. the thickness of the tail in the distribution. This family of distributions comprises the particular case of Gaussian with $\alpha = 2$. The second-order and higher-order statistics of the symmetric alpha-stable distribution ($\alpha \neq 2$) are infinity. μ is the location parameter (and assumed to be zero here). γ is the dispersion of the distribution and is similar to the variance of Gaussian random variable. In practice, the signal of semi-conducting electrical devices in communication and radar systems is subject to internal thermal Gaussian noises. Hence, a sum of independent $S\alpha S$ and Gaussian random process appears in a variety of practical situations mentioned above, namely, a $S\alpha SG$ distribution (Brcich & Zoubir, 1999; Ilow, Hatzinakos, & Venetsanopoulos, 1998; Samorodnitsky & Taqqu, 1996; Zha, 2007). The process is easily presented in the characteristic function

$$\phi(t) = \exp(-\gamma_{S\alpha S}|t|^{\alpha} - \gamma_G|t|^{\alpha}) \tag{49}$$

where $\gamma_{S\alpha S} > 0$ and $\gamma_G = \sigma_G^2/2 > 0$ are the dispersions of $S\alpha S$ and Gaussian random variables. σ_G^2 is related to the variance of the Gaussian component.

In order to effectively illustrate the good performance of RLMP-ELM algorithm, Gaussian and non-Gaussian datasets are considered in the study. For Gaussian dataset, Gaussian noises are added to the noise free training set or real data to generate training samples, called as Gaussian training set. For non-Gaussian dataset, Symmetry alpha-stable ($S\alpha S$) noise, The sum of independent $S\alpha S$ and Gaussian random noise ($S\alpha SG$), and Uniform noises are used to create training samples, called as $S\alpha S$ training set, $S\alpha SG$ training set and Uniform training set, respectively. Furthermore, all the simulations are carried out in MATLAB R2013a environment running in an Intel(R) Xeon(R) CPU, 3.50 GHz. The details of validation process are shown in the following sections.

4.1. SinC

In this section, a 'SinC' example is presented to confirm the theoretical analysis of the proposed RLMP-ELM algorithm. Here, the 'SinC' function is given as follows:

$$y(x) = \begin{cases} \sin(x)/x & x \neq 0 \\ 0 & x = 0 \end{cases} \tag{50}$$

5000 data are created for the training and validation set, respectively, where the input x are uniformly randomly distributed on the interval $(-10, 10)$.

For each type of dataset, we make model selection procedure firstly to determine the optimal architecture of the SLFN, that is the number of the hidden nodes. Then, we illustrate the performance of RLMP-ELM algorithm by comparing with ELM and OS-ELM algorithms.

4.1.1. Model selection

The estimation of optimal architecture of the network is called model selection in the literature. It is problem specific and has to be predetermined. For RLMP-ELM, ELM and OS-ELM algorithms, the optimal number of hidden units needs to be determined.

For Gaussian training set, the model selection procedure is shown as follows. For RLMP-ELM algorithm, the optimal p value is selected from the range [1.1, 2] with the interval 0.1. For ELM or OS-ELM algorithms, the training process is performed with different number of hidden nodes which is chosen from the range [2, 50] with the interval 2. Here, Monte Carlo method is used and over 200 trials are conducted for each number of hidden nodes. In each trial, random zero mean Gaussian noises with variance 0.16 are created and added to all training samples to generate the Gaussian training set. After each trial, the testing set without any noises is used to validate the performance of the algorithm. The average performance is calculated after over 200 trials and is shown in the Fig. 1(a). The Root Mean Square Error (RMSE) of the testing set is used as the criterion of the algorithm's performance.

In Fig. 1(a), the top two curves correspond to validation error for RLMP-ELM with $p = 1.6$ and $p = 1.2$ and the bottom three curves correspond to validation error for ELM, OS-ELM and RLMP-ELM with $p = 2$. The performance of the bottom three curves are similar. The Gaussian activation function is selected here for the hidden nodes. As can be observed from the figure, the lowest validation error is achieved when the number of hidden nodes of these algorithms is within the range [14, 24]. Therefore, one can choose the optimal hidden unit numbers for all the algorithms (in 'SinC' case) from the range. It can also be seen that RMSE curves for all algorithms are smooth. It implies that all algorithms are not sensitive to the network size.

For $S\alpha S$ and $S\alpha SG$ training set, the model selection procedures are the same as that of Gaussian training set. But, different types of noises are added on the training samples to create corresponding training set as mentioned above. For $S\alpha S$ training set, Symmetry alpha-stable random noise ($\alpha = 1.2$ and the dispersion $\gamma_{S\alpha S} = 0.02$) are used. For $S\alpha SG$ training set, the sum of independent $S\alpha S$ ($\alpha = 1.2$, $\gamma_{S\alpha S} = 0.02$) and Gaussian (zero mean, the variance is 0.16) random noises are used. The performances of all algorithms based on these two training set are shown in Fig. 1(b) and (c), respectively. In these two figures, the lowest validation error is achieved when the number of hidden nodes of ELM algorithm and the rest algorithms are within the ranges [10, 20] and [14, 24]. Furthermore, the RMSE curves of RLMP-ELM algorithms with $p = 1.2$ and $p = 1.6$ are quite smooth compared to ELM, OS-ELM and RLMP-ELM algorithm with $p = 2$. It implies that RLMP-ELM algorithms with $p = 1.2$ and $p = 1.6$ are not sensitive to the network size although the training samples are disturbed by the Symmetry alpha-stable random noise.

The same model selection procedures are used for Uniform training set and large uniform noise distributed in $[-0.4, 0.4]$ has been added to all the training samples. Different from above three datasets, for RLMP-ELM algorithm, the value of p is selected from the range [2, 4] with the interval 0.5. The performances of all algorithms are shown in Fig. 1(d). In the figure, the five curves almost coincide with each other. As it can be seen from the figure, the algorithms with the number of hidden nodes within [18, 26] can obtain the lowest validation error. In the same time, the RMSE curves of all algorithms are smooth and it is shown that these algorithms are all not sensitive to the network size while the training samples are disturbed by the Uniform random noise.

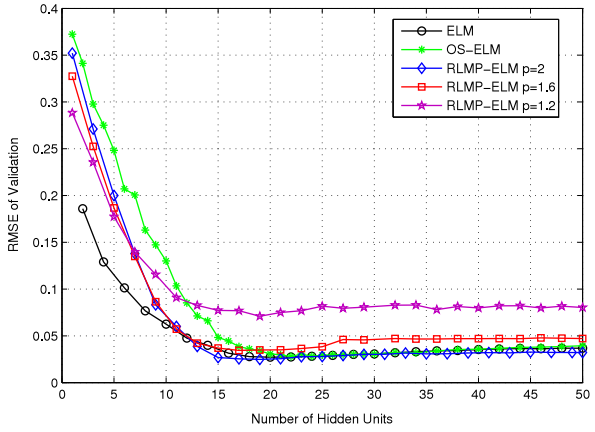
4.1.2. Performance evaluation of RLMP-ELM algorithm

In this part, the performance of RLMP-ELM algorithms with different p values is discussed. According to the analysis above, 20 is selected as the optimal number of hidden nodes for RLMP-ELM, ELM and OS-ELM algorithms.

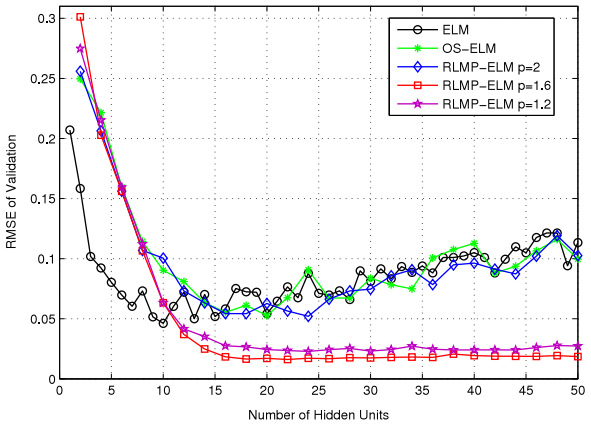
1. Convergence performance

Primarily, we compare the online learning processes of RLMP-ELM and OS-ELM algorithms. The convergence curves in terms of the validation RMSE are illustrated in Fig. 2 for different training sets. Each curve in these figures is the averaged result over 200 independent trials.

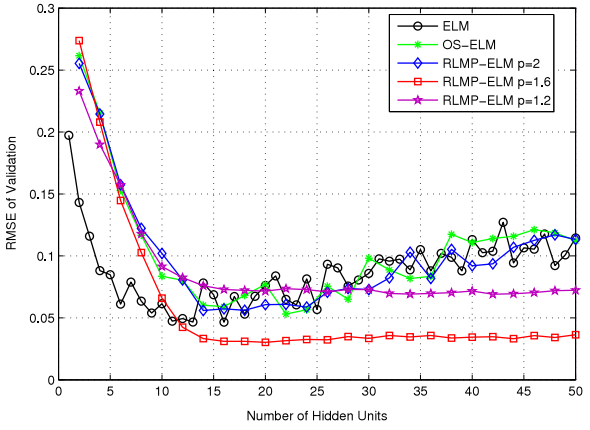
As can be shown from Fig. 2(a), all algorithms are robust to the Gaussian noises. Apart from RLMP-ELM algorithm with $p = 1.2$, other algorithms almost have the same convergence rate and



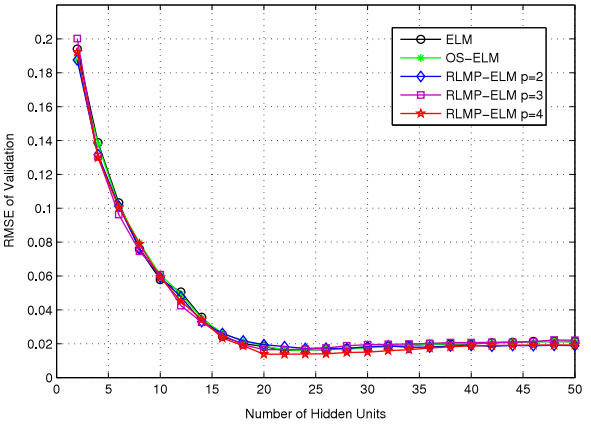
(a) Gaussian training set.



(b) $S\alpha S$ training set.



(c) $S\alpha SG$ training set.



(d) Uniform training set.

Fig. 1. Model selection for 'SinC' based on four types of training sets.

the stable testing performance. The convergence rate and stable performance of RLMP-ELM algorithm with $p = 1.2$ are not as well as that of other algorithms.

Fig. 2(b) illustrates that both RLMP-ELM algorithms with $p = 1.2$ and $p = 1.6$ are robust to the Symmetry alpha-stable random noises so that the curves remain stable. Furthermore, these two curves are below other two curves so that the stable testing RMSE of the corresponding algorithms are lower. For the second-order statistics of the Symmetry alpha-stable random noises is infinity, RLMP-ELM algorithm with $p = 2$ and OS-ELM algorithm are sensitive to the Symmetry alpha-stable random noises and stable performances are not as well as that of the other two algorithms. Moreover, the convergence rates of RLMP-ELM algorithms with $p = 1.2$ and $p = 1.6$ are faster than that of RLMP-ELM algorithm with $p = 2$ and OS-ELM algorithm.

It is clear from Fig. 2(c) that both RLMP-ELM algorithms with $p = 1.2$ and $p = 1.6$ are robust to the Symmetry alpha-stable random noise and Gaussian noise, but the stable performance of the algorithm with $p = 1.2$ is worse than that of the algorithm with $p = 1.6$ since there are Gaussian random noises. RLMP-ELM algorithm with $p = 2$ and OS-ELM algorithm are still sensitive to the noises because there are the Symmetry alpha-stable random noises. The convergence rate of RLMP-ELM algorithm with $p = 1.6$ is the fastest than that of the other algorithms.

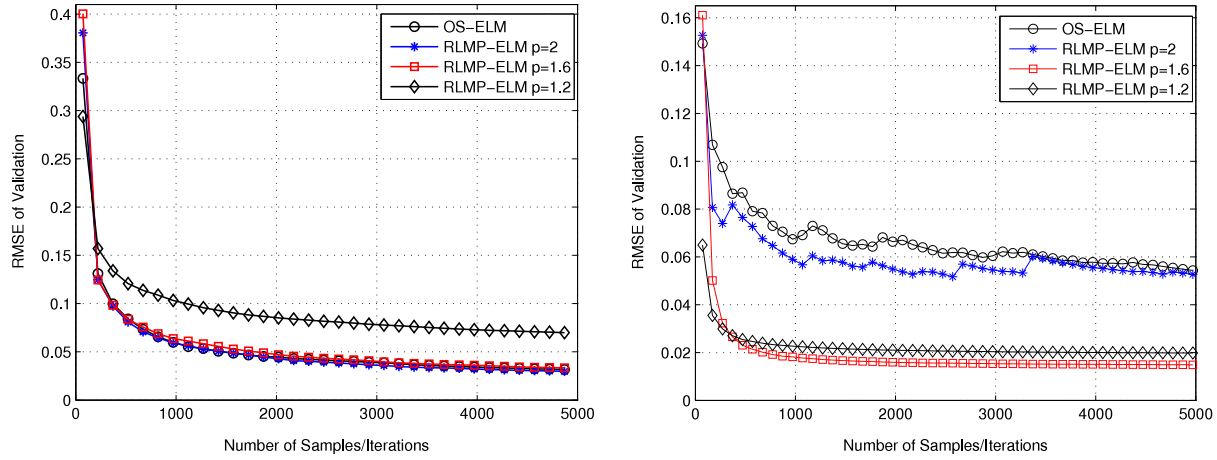
In Fig. 2(d), it can be seen that all algorithms are robust to the Uniform noises and the curves remain steady. Comparing with other algorithms, RLMP-ELM algorithm with $p = 4$ has the best convergence performance. Other algorithms almost have the same convergence rate and the stable testing performance. Here, the p value is usually bounded above by a certain positive number. In our simulations, the algorithm can sometimes be unstable, while the p value is greater than 6.

2. Details of performance

Furthermore, the more details of the comparison about RLMP-ELM algorithm with different p values, ELM and OS-ELM algorithms are summarized in the follow-up table. The averaged results over 200 independent trials on each algorithm in terms of the running time, the RMSE and the variance of the RMSE of the training and testing process and the number of hidden nodes are presented in Table 1.

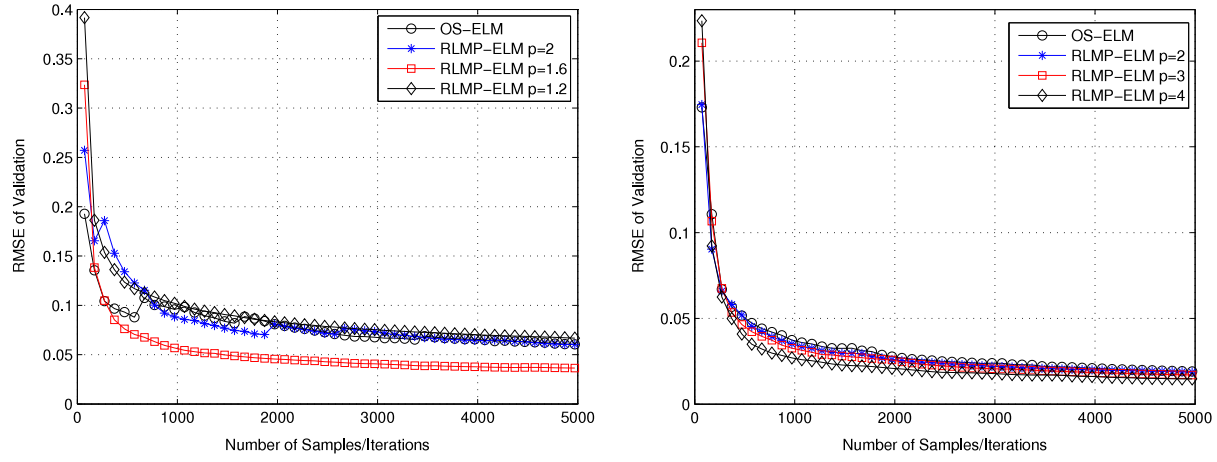
As observed from Table 1, the performances of RLMP-ELM with different p, ELM and OS-ELM based on Gaussian training dataset are similar to each other. All algorithms are robust to the Gaussian distribution data. There is only one obvious difference that the training time taken by ELM is much less than that taken by other algorithms. Just as the above analysis, the computation complexity of ELM, OS-ELM and RLMP-ELM algorithms is the same, but it costs more running time in the last two algorithms for conducting data one by one.

The performances of all algorithms according to $S\alpha S$ training dataset are also shown in Table 1. The validation RMSE of RLMP-ELM algorithm with p in the range of $[1.2, 1.8]$ is much better than that of other algorithms. However, out of all learning algorithms, RLMP-ELM algorithm with $p = 1.6$ obtains the lowest testing root-mean-square error (RMSE) 0.0164 while the criteria of ELM and OS-ELM are both above 0.05. In conclusion, the algorithms with



(a) Gaussian training set.

(b) $S\alpha S$ training set.



(c) $S\alpha SG$ training set.

(d) Uniform training set.

Fig. 2. Learning evolution for 'SinC' based on four types of training sets.

least mean square criterion are sensitive to the data with Impulsive characteristic, while RLMP-ELM with p value in range [1.2, 1.8] is more robust to Impulsive training data used here.

Table 1 illustrates the performances of all algorithms based on $S\alpha SG$ training dataset, too. The validation RMSE of RLMP-ELM algorithm with p value in the range of [1.4, 1.8] is a little better than that of other algorithms. RLMP-ELM algorithm with $p = 1.6$ still obtains the lowest testing RMSE 0.0349. But, the value of this criterion is only a little less than the performances of ELM and OS-ELM algorithms for the sum of Impulsive and Gaussian data. The performance of RLMP-ELM algorithm with $p = 1.2$ is worse than all the other algorithms because of the Gaussian random noises.

Finally, the performances of all algorithms based on Uniform training dataset are shown in the bottom of Table 1. The validation RMSE of RLMP-ELM algorithm with p value in the range of [3, 4] is slightly better than that of other algorithms. RLMP-ELM algorithm with $p = 4$ obtains the lowest testing RMSE because the Uniform data are bounded.

From the simulation results of 'SinC' case, we have observed that RLMP-ELM algorithm with appropriate p value can obtain better performance on non-Gaussian dataset than ELM and OS-ELM algorithms can do. In order to further illustrate the good performance of the proposed algorithm, we have conducted the detailed simulation test on the two real world datasets and non-stationary time-series prediction problem in the following sections.

4.2. Regression benchmark: Airfoil self-noise problem

For Airfoil self-noise problem, 50% and 50% samples are randomly chosen for training and testing at each trial. Thus, the number of training data and testing data is the same, 751. The training dataset is added on four different types of random noises. Considering the statistical property of the random noises, 5000 random noise samples are created and 751 noisy samples are randomly chosen from them at each trail. Then, 751 noisy samples are added on training data to generate training dataset. According to the model selection work presented above, 20 is selected as the optimum number of hidden units. For each type of training dataset, the average results over 200 trails are shown in Table 2.

The detailed performances of each algorithm for Gaussian training dataset are illustrated in Table 2. As can be observed from the table, all algorithms obtain similar performance. Only the running time taken by the ELM algorithm is much less than that taken by other algorithms. This is because the ELM algorithm processes all samples in one time and the other two algorithms process the samples one by one.

As can be observed from Table 2, in case of $S\alpha S$ training dataset, the testing RMSE of RLMP-ELM with p value in the range of [1.2, 1.8] is less than that of other three algorithms, ELM, OS-ELM and RLMP-ELM algorithm with $p = 2$. The lowest testing RMSE obtained by the RLMP-ELM algorithm with $p = 1.6$ is about half of those of ELM, OS-ELM and RLMP-ELM algorithm with $p = 2$.

Table 1
Performance comparison of RLMP-ELM, ELM and OS-ELM algorithms for 'SinC' case based on four types of training sets.

Noise type	Algorithms		Training			Validation			#nodes
			RMSE	Dev	Time (s)	RMSE	Dev	Time (s)	
Gauss	ELM		0.3989	0.0037	0.0108	0.0273	0.0055	0.0013	20
	OS-ELM		0.3996	0.0048	0.6435	0.0307	0.0095	0.0051	20
	RLMP-ELM	$p = 1.6$	0.3994	0.0072	0.6805	0.0311	0.0078	0.0206	20
		$p = 1.8$	0.3994	0.0039	0.6760	0.0268	0.0066	0.0011	20
		$p = 2.0$	0.3988	0.0038	0.6804	0.0259	0.0041	0.0098	20
		$p = 2.2$	0.3999	0.0053	0.6416	0.0271	0.0055	0.0077	20
		$p = 2.4$	0.3994	0.0039	0.6645	0.0270	0.0054	0.0030	20
$S\alpha S$	ELM		0.8325	0.1092	0.0119	0.0534	0.0114	0.0039	20
	OS-ELM		0.7973	0.1223	0.6371	0.0566	0.0090	0.0049	20
	RLMP-ELM	$p = 1.2$	0.8371	0.1121	0.6232	0.0211	0.0043	0.0022	20
		$p = 1.4$	0.8575	0.1070	0.6148	0.0164	0.0034	0.0033	20
		$p = 1.6$	0.8469	0.1078	0.6229	0.0161	0.0039	0.0052	20
		$p = 1.8$	0.8269	0.1192	0.6222	0.0240	0.0051	0.0049	20
		$p = 2$	0.8430	0.1150	0.6264	0.0536	0.0117	0.0062	20
$S\alpha SG$	ELM		0.8827	0.0801	0.0101	0.0593	0.0110	0.0036	20
	OS-ELM		0.8624	0.0550	0.6102	0.0593	0.0073	0.0047	20
	RLMP-ELM	$p = 1.2$	0.8793	0.0648	0.6170	0.0701	0.0145	0.0040	20
		$p = 1.4$	0.8912	0.0812	0.6216	0.0478	0.0112	0.0095	20
		$p = 1.6$	0.8734	0.0907	0.6126	0.0349	0.0068	0.0039	20
		$p = 1.8$	0.8841	0.0951	0.6178	0.0420	0.0069	0.0035	20
		$p = 2$	0.8359	0.0900	0.6195	0.0532	0.0092	0.0021	20
Uniform	ELM		0.2307	0.0016	0.0124	0.0172	0.0049	0.0021	20
	OS-ELM		0.2314	0.0023	0.6398	0.0189	0.0092	0.0044	20
	RLMP-ELM	$p = 2.0$	0.2306	0.0016	0.6815	0.0183	0.0071	0.0200	20
		$p = 3.0$	0.2311	0.0019	0.6359	0.0170	0.0087	0.0156	20
		$p = 4.0$	0.2312	0.0024	0.6532	0.0147	0.0088	0.0192	20

Table 2
Performance comparison of RLMP-ELM, ELM and OS-ELM algorithms for 'Airfoil Self-Noise' case based on four types of training sets.

Noise type	Algorithms		Training			Validation			#nodes
			RMSE	Dev	Time (s)	RMSE	Dev	Time (s)	
Gauss	ELM		0.3921	0.0021	0.0030	0.1310	0.0046	0.0011	20
	OS-ELM		0.4106	0.0108	0.0959	0.1305	0.0079	0.0019	20
	RLMP-ELM	$p = 1.6$	0.4124	0.0109	0.0950	0.1400	0.0086	0.0021	20
		$p = 1.8$	0.4105	0.0100	0.1049	0.1315	0.0063	0.0016	20
		$p = 2.0$	0.4086	0.0117	0.0960	0.1299	0.0055	0.0013	20
		$p = 2.2$	0.4103	0.0108	0.0952	0.1313	0.0063	0.0017	20
		$p = 2.4$	0.4110	0.0118	0.0958	0.1339	0.0065	0.0011	20
$S\alpha S$	ELM		1.1407	0.8213	0.0031	0.2231	0.1057	0.0014	20
	OS-ELM		1.1280	0.8031	0.0944	0.2187	0.1084	0.0016	20
	RLMP-ELM	$p = 1.2$	1.1109	0.9141	0.0957	0.1373	0.0126	0.0020	20
		$p = 1.4$	1.2117	0.8255	0.0957	0.1261	0.0070	0.0011	20
		$p = 1.6$	1.2564	0.9750	0.0964	0.1252	0.0095	0.0012	20
		$p = 1.8$	1.1772	0.7701	0.0968	0.1505	0.0594	0.0014	20
		$p = 2$	1.1232	0.8171	0.1051	0.2278	0.1282	0.0018	20
$S\alpha SG$	ELM		1.2437	0.8234	0.0031	0.2324	0.1171	0.0012	20
	OS-ELM		1.2566	0.8455	0.1054	0.2304	0.1253	0.0013	20
	RLMP-ELM	$p = 1.2$	1.2356	0.7851	0.0953	0.2270	0.0460	0.0017	20
		$p = 1.4$	1.3194	0.8560	0.0958	0.1798	0.0600	0.0016	20
		$p = 1.6$	1.1721	0.7553	0.0966	0.1559	0.0149	0.0016	20
		$p = 1.8$	1.2587	0.8088	0.0960	0.1658	0.0369	0.0023	20
		$p = 2$	1.2418	0.7398	0.1051	0.2414	0.1287	0.0017	20
Uniform	ELM		0.0024	0.0015	0.2642	0.0073	0.1370	0.0097	20
	OS-ELM		0.0969	0.0019	0.2649	0.0071	0.1362	0.0095	20
	RLMP-ELM	$p = 2.0$	0.0968	0.0022	0.2650	0.0073	0.1370	0.0084	20
		$p = 3.0$	0.0981	0.0020	0.2656	0.0055	0.1353	0.0051	20
		$p = 4.0$	0.0984	0.0019	0.2616	0.0063	0.1311	0.0068	20

Table 2 shows the performances of all algorithms according to the $S\alpha SG$ training dataset. The testing RMSE of RLMP-ELM with p in the range of $[1.4, 1.8]$ is a little less than that of other algorithms since there are Gaussian random noises. The lowest testing RMSE is obtained by RLMP-ELM algorithm with $p = 1.6$, which is as $S\alpha S$ training dataset.

For Uniform training dataset, the performances of RLMP-ELM with p in the range of $[3, 4]$ are a little less than those of other algorithms since there are Uniform data. The lowest testing RMSE is obtained by RLMP-ELM algorithm with $p = 4$. The details are illustrated in Table 2.

Table 3
Performance comparison of RLMP-ELM, ELM and OS-ELM algorithms for 'Yacht Hydrodynamics' case based on four types of training sets.

Noise type	Algorithms		Training			Validation			#nodes
			RMSE	Dev	Time (s)	RMSE	Dev	Time (s)	
Gauss	ELM		0.4244	0.0262	0.0002	0.1758	0.0210	0.0001	10
	OS-ELM		0.4236	0.0266	0.0052	0.1759	0.0233	0.0001	10
	RLMP-ELM	$p = 1.4$	0.4382	0.0266	0.0075	0.1676	0.0247	0.0001	10
		$p = 1.6$	0.4328	0.0264	0.0059	0.1624	0.0213	0.0002	10
		$p = 1.8$	0.4276	0.0251	0.0055	0.1672	0.0214	0.0001	10
		$p = 2.0$	0.4248	0.0285	0.0053	0.1788	0.0235	0.0002	10
		$p = 2.2$	0.4250	0.0257	0.0080	0.1886	0.0243	0.0003	10
SαS	ELM		0.8583	0.4223	0.0006	0.2783	0.1787	0.0001	10
	OS-ELM		0.8158	0.4753	0.0054	0.2711	0.1762	0.0003	10
	RLMP-ELM	$p = 1.2$	0.8604	0.3608	0.0055	0.1257	0.0207	0.0002	10
		$p = 1.4$	0.8441	0.3434	0.0060	0.1276	0.0212	0.0002	10
		$p = 1.6$	0.8381	0.3541	0.0056	0.1470	0.0396	0.0003	10
		$p = 1.8$	0.8687	0.3611	0.0062	0.2088	0.1316	0.0001	10
		$p = 2$	0.7984	0.3594	0.0054	0.2605	0.1673	0.0001	10
SαSG	ELM		0.7464	0.3858	0.0002	0.3044	0.1392	0.0002	10
	OS-ELM		0.7765	0.4206	0.0057	0.3190	0.1463	0.0001	10
	RLMP-ELM	$p = 1.2$	0.8049	0.4551	0.0052	0.2118	0.0369	0.0002	10
		$p = 1.4$	0.8173	0.4053	0.0062	0.2073	0.0385	0.0003	10
		$p = 1.6$	0.7935	0.4026	0.0067	0.2113	0.0481	0.0001	10
		$p = 1.8$	0.7345	0.4106	0.0057	0.2449	0.0801	0.0001	10
		$p = 2$	0.7513	0.4662	0.0045	0.3112	0.1476	0.0002	10
Uniform	ELM		0.2872	0.0173	0.0002	0.1557	0.0140	0.0001	10
	OS-ELM		0.2845	0.0183	0.0069	0.1559	0.0155	0.0001	10
	RLMP-ELM	$p = 2.0$	0.2865	0.0184	0.0058	0.1570	0.0190	0.0001	10
		$p = 3.0$	0.2850	0.0179	0.0061	0.1519	0.0107	0.0001	10
		$p = 4.0$	0.2844	0.0149	0.0059	0.1481	0.0102	0.0001	10

4.3. Regression benchmark: Yacht hydrodynamics problem

As done in the case of 'airfoil self-noise' problem, 50% and 50% samples of Yacht hydrodynamics dataset are randomly chosen for training and testing at each trial. The procedure of creating training dataset is entirely the same as that in the 'airfoil self-noise' case. According to the model selection procedure, 10 is selected as the optimum number of hidden units. For each type of training dataset, the average results over 200 trails are shown in Table 3.

For this problem, the performances of all algorithms based on different types of training dataset are similar with that in the 'airfoil self-noise' case. For Gaussian training dataset, the performances of all algorithms are substantially similar, as observed from Table 3. From the table, it can be seen that RLMP-ELM with p value in the range of $[1.2, 1.8]$ has the less values of the testing RMSE in case of $S\alpha S$ and $S\alpha SG$ training dataset. The lowest testing RMSE is obtained by RLMP-ELM with $p = 1.4$ in both of these two training datasets. The performance details of all the algorithms for Uniform training set are illustrated in the bottom of Table 3. From the table, it can be seen that RLMP-ELM algorithm with p value in the range of $[3, 4]$ has little less values of the testing RMSE and the lowest testing RMSE is achieved by the algorithm with $p = 4$.

4.4. Time-series prediction problems

Nonlinear time series prediction arises in the development of techniques for dynamic systems modeling, that is the basis of many real-world problems such as detecting arrhythmia in heartbeats, stock market indices, etc. One of the classical benchmark problems in the literature is the non-stationary Mackey–Glass chaotic time series generated by the following differential delay equation:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1 + x(t-\tau)^{10} - 0.1x(t)} \tag{51}$$

where $x(t)$ is the value of time series at time t. When $\tau = 17$, $x(0) = 1.2$, and $x(t) = 0$ for $t < 0$, a non-periodic and non-convergent chaotic time series is obtained. The time series is conducted using the fourth-order Runge–Kutta method with a step size of 0.1. The time series is predicted with $v = 50$ sample steps ahead using the four past samples: s_{n-v}, s_{n-v-6}, s_{n-v-12} and s_{n-v-18}. Hence, the nth input–output instance is

$$X_n = [s_{n-v}\ s_{n-v-6}\ s_{n-v-12}\ s_{n-v-18}]^T$$
$$Y_n = s_n.$$

In this simulation, the number of training observation samples is 4000 and the time t is from 1 to 400. The number of testing observation samples is 500 and the time t is from 401 to 450.

The same Gaussian and non-Gaussian noises as described above are added to the 4000 training data in each trial. Besides, uniform noise distributed in range $[-0.4, 0.4]$ is added to the free noise training data.

The performance comparisons of ELM, OS-ELM and RLMP-ELM algorithm with different p values refer to four kinds of training dataset are shown in Table 4. In summary, in case that the impulsive property of measurement noise is strong, the better performance can be obtained by RLMP-ELM with p in the range of $[1.2, 1.8]$. For only Gaussian random measurement noises, the performance of RLMP-ELM algorithm with p in the range of $[1.4, 2.4]$ are similar with that of ELM and OS-ELM algorithms. In bounded uniform noise case, the RLMP-ELM algorithm with $p = 4$ has got the lowest testing RMSE, just as shown in the above regression benchmarks.

5. Conclusion

This paper proposes an efficient and accurate online sequential learning algorithm for single-hidden layer feedforward neural networks (SLFNs) called recursive least mean p-power extreme learning machine (RLMP-ELM). The property of activation functions for hidden units here is the same as those for ELM and OS-ELM algorithms, which can be any bounded nonconstant piecewise

Table 4
Performance comparison of RLMP-ELM, ELM and OS-ELM algorithms for 'Mackey–Glass' case based on four types of training sets.

Noise type	Algorithms		Training			Validation			#nodes
			RMSE	Dev	Time (s)	RMSE	Dev	Time (s)	
Gauss	ELM		0.4111	0.0051	0.0174	0.1042	0.0048	0.0005	20
	OS-ELM		0.4106	0.0068	0.5716	0.1028	0.0143	0.0027	20
	RLMP-ELM	$p = 1.4$	0.4110	0.0043	0.5734	0.1029	0.0039	0.0016	20
		$p = 1.6$	0.4107	0.0035	0.5878	0.1015	0.0040	0.0021	20
		$p = 1.8$	0.4103	0.0034	0.5744	0.1017	0.0039	0.0012	20
		$p = 2.0$	0.4102	0.0034	0.5766	0.1019	0.0041	0.0012	20
		$p = 2.2$	0.4100	0.0034	0.5703	0.1025	0.0038	0.0019	20
$S\alpha S$	ELM		1.5440	0.8340	0.0195	0.1526	0.0488	0.0014	20
	OS-ELM		1.7575	0.9546	0.6040	0.1630	0.0605	0.0030	20
	RLMP-ELM	$p = 1.2$	1.5765	0.9711	0.5509	0.1023	0.0048	0.0016	20
		$p = 1.4$	1.6090	0.9549	0.5679	0.0999	0.0068	0.0016	20
		$p = 1.6$	1.7519	0.9434	0.6087	0.0998	0.0038	0.0012	20
		$p = 1.8$	1.7687	0.9762	0.5845	0.1087	0.0102	0.0008	20
		$p = 2$	1.6694	0.9541	0.5643	0.1600	0.0628	0.0005	20
$S\alpha SG$	ELM		1.8054	0.9970	0.0225	0.1674	0.0613	0.0010	20
	OS-ELM		1.7408	0.8900	0.5988	0.1606	0.0539	0.0026	20
	RLMP-ELM	$p = 1.2$	1.6780	0.9466	0.5665	0.1028	0.0061	0.0019	20
		$p = 1.4$	1.8138	0.9993	0.5726	0.0995	0.0038	0.0009	20
		$p = 1.6$	1.6854	0.8785	0.5663	0.0999	0.0036	0.0016	20
		$p = 1.8$	1.7463	0.9493	0.5728	0.1094	0.0119	0.0014	20
		$p = 2$	1.6940	0.9717	0.5444	0.1619	0.0674	0.0017	20
Uniform	ELM		0.2511	0.0031	0.0219	0.1019	0.0043	0.0031	20
	OS-ELM		0.2505	0.0026	0.5664	0.1011	0.0045	0.0036	20
	RLMP-ELM	$p = 2.0$	0.2511	0.0026	0.5455	0.1016	0.0043	0.0022	20
		$p = 3.0$	0.2520	0.0028	0.5721	0.0977	0.0046	0.0037	20
		$p = 4.0$	0.2501	0.0027	0.5686	0.0971	0.0050	0.0020	20

continuous function for additive nodes and any integrable piecewise continuous functions for RBF nodes. The RLMP-ELM algorithm maintains the computationally simple extreme learning machine architecture but a least mean p-power (LMP) error criterion aiming to minimize the p powers of the error provides a mechanism to update the output weights sequentially. under the same architecture, RLMP-ELM has the same computational complexity as that of ELM and OS-ELM. In order to show the effectiveness and good performance of the proposed method, a comparison with different p values, ELM and the OS-ELM algorithm has been performed under the real world benchmark regression and non-stationary time-series prediction problems. The results show that the proposed RLMP-ELM can obtain better performance in non-Gaussian situations than ELM and OS-ELM algorithms. Furthermore, the proposed algorithm has several interesting and significant features:

1. For Gaussian distributed data, the RLMP-ELM algorithm with p ($p = 2$) can achieve better generalization performance and more accurate results.
2. For non-Gaussian heavy-tailed distributed data, the RLMP-ELM algorithm with p ($1 \leq p < 2$) can obtain better generalization performance and more accurate results.
3. As for non-Gaussian light-tailed distributed data, the RLMP-ELM algorithm with p ($2 < p \leq 4$) can get better generalization performance and more accurate results, especially ($p = 4$).

It should be worth pointing out that there is a point which requires more work. Although we obtain the range of p values for heavy-tailed distribution and light-tailed distribution process, how to determine the exact value of p, with which the RLMP-ELM has the best generalization performance and most accurate result, is not very clear. We will discuss it in our future works.

Acknowledgments

This work is funded in part by National Natural Science Foundation of China (grant number 61403300, 91648208), 973 Program (grant number 2015CB351703), National Science Council of ShaanXi Province (grant number 2014JM8337) and the Fundamental Research Funds for the Central Universities.

References

Beaulieu, N.C., & Niranjayan, S. (2007). New uwb receiver designs based on a gaussian-laplacian noise-plus-mai model. In *2007 IEEE international conference on communications* (pp. 4128–4133).

Bhotto, M. Z. A., & Antoniou, A. (2011). Robust recursive least-squares adaptive-filtering algorithm for impulsive-noise environments. *Signal Processing Letters IEEE*, *18*(3), 185–188.

Bouvet, M., & Schwartz, S. C. (1989). Comparison of adaptive and robust receivers for signal detection in ambient underwater noise. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*(5), 621–626.

Brcich, R., & Zoubir, A. (1999). Estimation and detection in a mixture of symmetric alpha stable and gaussian interference. In *Higher-order statistics, 1999. Proceedings of the IEEE signal processing workshop on* (pp. 219–223).

Chambers, J. A., Tanrikulu, O., & Constantinides, A. G. (1994). Least mean mixed-norm adaptive filtering. *Electronics Letters*, *30*(19), 1574–1575.

Chan, S. C., & Zou, Y. X. (2004). A recursive least m-estimate algorithm for robust adaptive filtering in impulsive noise: Fast algorithm and convergence performance analysis. *IEEE Transactions on Signal Processing*, *52*(4), 975–991.

Chang, P. C., & Fan, C. Y. (2008). A hybrid system integrating a wavelet and tsk fuzzy rules for stock price forecasting. *IEEE Transactions on Systems, Man and Cybernetics Part C*, *38*(6), 802–815.

Chen, B., Liu, X., Zhao, H., & Principe, J. C. (2017). Maximum correntropy kalman filter. *Automatica*, *76*, 70–77.

Chen, B., Wang, J., Zhao, H., & Zheng, N. (2015). Convergence of a fixed-point algorithm under maximum correntropy criterion. *IEEE Signal Processing Letters*, *22*(10), 1723–1727.

Chen, B., Xing, L., Liang, J., Zheng, N., & Príncipe, J. C. (2014). Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion. *IEEE Signal Processing Letters*, *21*(7), 880–884.

Chen, B., Xing, L., Wu, Z., Liang, J., Príncipe, J. C., & Zheng, N. (2015). Smoothed least mean p-power error criterion for adaptive filtering. *Digital Signal Processing*, *40*, 154–163.

Chen, B., Xing, L., Zhao, H., Zheng, N., & Príncipe, J. C. (2016). Generalized correntropy for robust adaptive filtering. *IEEE Transactions on Signal Processing*, *64*(13), 3376–3387.

Chen, B., Zhao, S., Zhu, P., & Príncipe, J. C. (2012). Quantized kernel least mean square algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(1), 22–32.

Chen, B., Zhao, S., Zhu, P., & Príncipe, J. C. (2013). Quantized kernel recursive least squares algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *24*(9), 1484–1491.

Chen, B., Zhu, Y., Hu, J., & Principe, J. C. (2013). *System parameter identification: information criteria and algorithms*. Elsevier.

Chrissan, D.A. (1998). Statistical analysis and modeling of low-frequency radio noise and improvement of low-frequency communications, Ph.D. dissertation, Elect. Eng. Dept., Stanford Univ.

Deng, W.-Y., Zheng, Q.-H., & Wang, Z.-M. (2014). Cross-person activity recognition using reduced kernel extreme learning machine. *Neural Networks*, *53*, 1–7.

Diniz, P. S. (2008). *Adaptive filtering: algorithms and practical implementation*. Kluwer Academic Publishers.

Ferrari, S., & Stengel, R. F. (2005). Smooth function approximation using neural networks. *IEEE Transactions on Neural Networks*, *16*(1), 24–38.

Golestaneh, F., Pinson, P., & Gooi, H. B. (2016). Very short-term nonparametric probabilistic forecasting of renewable energy generation - with application to solar energy. *IEEE Transactions on Power Systems*, 1–14.

Horata, P., Chiewchanwattana, S., & Sunat, K. (2013). Robust extreme learning machine. *Neurocomputing*, *102*, 31–44.

Hou, M., & Han, X. (2010). Constructive approximation to multivariate function by decay rbf neural network. *IEEE Transactions on Neural Networks*, *21*(9), 1517–1523.

Huang, G.-B., Chen, L., & Siew, C.-K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, *17*(4), 879–892.

Huang, G.-B., Liang, N.-Y., Rong, H.-J., Saratchandran, P., & Sundararajan, N. (2005). On-line sequential extreme learning machine. In *The IASTED international conference on computational intelligence, CI 2005*, Calgary, Canada.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theorey and applications. *Neurocomputing*, *70*(1), 489–501.

Ilow, J., Hatzinakos, D., & Venetsanopoulos, A. N. (1998). Performance of fh ss radio networks with interference modeled as a mixture of gaussian and alpha-stable noise. *IEEE Transactions on Communications*, *46*(4), 509–520.

Kismode, P. (2000). Alpha-stable distributions in signal processing of audio signals. In *The proceedings of the 41st conference of simulation model*, September (pp. 87–94).

LeCun, Y., Bottou, L., Orr, G. B., & Müller, K. R. (1998). Efficient BackProp. In *Neural networks: tricks of the trade* (pp. 9–50). Berlin, Heidelberg: Springer Berlin, Heidelberg, (Chapter).

Lee, J., & Tepedelenlioglu, C. (2014). Distributed detection in coexisting large-scale sensor networks. *IEEE Sensors Journal*, *14*(4), 1028–1034.

Li, Y., Jia, Z., & Li, X. (2014). Task scheduling based on weather forecast in energy harvesting sensor systems. *IEEE Sensors Journal*, *14*(14), 3763–3765.

Liang, N.-Y., Huang, G.-B., Saratchandran, P., & Sundararajan, N. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on Neural Networks*, *17*(6), 1411–1423.

Liao, S., & Chung, A. C. S. (2010). Feature based nonrigid brain mr image registration with symmetric alpha stable filters. *IEEE Transactions on Medical Imaging*, *29*(1), 106–119.

Lim, J.-s., Lee, S., & Pang, H.-S. (2013). Low complexity adaptive forgetting factor for online sequential extreme learning machine (os-elm) for application to nonstationary system estimations. *Neural Computing and Applications*, *22*(3), 569–576.

Ma, W., Qu, H., Zhao, J., Chen, B., & Gui, G. (2015). Sparsity aware normalized least mean p-power algorithms with correntropy induced metric penalty. In *2015 IEEE international conference on digital signal processing, DSP* (pp. 638–642).

MacKay, D. J. (1992). A practical Bayesian framework for backpropagation networks. *Neural Computation*, *4*(3), 448–472.

Man, Z., Lee, K., Wang, D., Cao, Z., & Khoo, S. (2012). Robust single-hidden layer feedforward network-based pattern classifier. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(12), 1974–1986.

Man, Z., Lee, K., Wang, D., Cao, Z., & Miao, C. (2011). A new robust training algorithm for a class of single-hidden layer feedforward neural networks. *Neurocomputing*, *74*(16), 2491–2501.

Mao, G. (2005). A timescale decomposition approach to network traffic prediction. *IEICE Transactions on Communications*, *E88B*(10), 3974–3981.

Matias, T., Gabriel, D., Souza, F., Araújo, R., & Pereira, J.C. (2013). Fault detection and replacement of a temperature sensor in a cement rotary kiln. In *2013 IEEE 18th conference on emerging technologies factory automation, ETFA* (pp. 1–8).

Meir, R., & Maiorov, V. E. (2000). On the optimality of neural-network approximation using incremental algorithms. *IEEE Transactions on Neural Networks*, *11*(2), 323–337.

Nikias, C. L., & Shao, M. (1995). *Signal processing with alpha-stable distributions and applications*. New York, NY, USA: Wiley-Interscience.

Pei, S.-C., & Tseng, C.-C. (1994a). Least mean p-power error criterion for adaptive FIR filter. *IEEE Journal on Selected Areas in Communications*, *12*(9), 1540–1547.

Pei, S., & Tseng, C. (1994b). Least mean p-power error criterion for adaptive FIR filter. *IEEE Journal on Selected Areas in Communications*, *12*(9), 1540–1547.

Qin, P., Nishii, R., & Yang, Z. J. (2012). Selection of narx models estimated using weighted least squares method via gic-based method and l 1-norm regularization methods. *Nonlinear Dynamics*, *70*(3), 1831–1846.

Rong, H. J., Huang, G. B., Sundararajan, N., & Saratchandran, P. (2009). Online sequential fuzzy extreme learning machine for function approximation and classification problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *39*(4), 1067–1072.

Samorodnitsky, G., & Taqqu, M. S. (1996). Stable non-gaussian random processes: Stochastic models with infinite variance. *Journal of the American Statistical Association*, *90*(430).

Sapankevych, N. I., & Sankar, R. (2009). Time series prediction using support vector machines: A survey. *IEEE Computational Intelligence Magazine*, *4*(2), 24–38.

Shao, M., & Nikias, C. L. (1993). Signal processing with fractional lower order moments: stable processes and their applications. *Proceedings of the IEEE*, *81*(7), 986–1010.

Shi, Z., & Han, M. (2009). $\gamma-$C plane and robustness in static reservoir for nonlinear regression estimation. *Neurocomputing*, *72*(7–9), 1732–1743.

Soares, S. G., & Araújo, R. (2016). An adaptive ensemble of on-line extreme learning machines with variable forgetting factor for dynamic system prediction. *Neurocomputing*, *171*, 693–707.

Sun, S., Zhang, C., & Yu, G. (2006). A Bayesian network approach to traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, *7*(1), 124–132.

Trafalis, T.B., & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting, 6, 348–353.

Walach, E., & Widrow, B. (1984). The least mean fourth (lmf) adaptive algorithm and its family. *IEEE Transactions on Information Theory*, *30*(2), 275–283.

Wang, X., & Han, M. (2014). Online sequential extreme learning machine with kernels for nonstationary time series prediction. *Neurocomputing*, *145*, 90–97.

Wen, F. (2013). Diffusion least-mean p-power algorithms for distributed estimation in alpha-stable noise environments. *Electronics Letters*, *49*(21), 1355–1356.

Xiao, Y. T. Y., & Shida, K. (1999). Adaptive algorithm based on least mean p-power error criterion for fourier analysis in additive noise. *IEEE Transactions on Signal Processing*, *47*(4), 1172–1181.

Xiao, Y., Tadokoro, Y., & Shida, K. (1999). Adaptive algorithm based on least mean p-power error criterion for fourier analysis in additive noise. *IEEE Transactions on Signal Processing*, *47*(4), 1172–1181.

Yang, J., Chen, P., Rong, H.-J., & Chen, B. (2016). Least mean p-power extreme learning machine for obstacle avoidance of a mobile robot. In *The international joint conference on neural networks, IJCNN*, Vancouver, Canada.

Yang, J., Shi, Y., & Rong, H.-J. (2016). Random neural q-learning for obstacle avoidance of a mobile robot in unknown environments. *Advances in Mechanical Engineering*, *8*(7).

Ye, Y., Squartini, S., & Piazza, F. (2013). Online sequential extreme learning machine in nonstationary environments. *Neurocomputing*, *116*, 94–101.

Zha, D. (2006). Robust multiuser detection method based on least p -norm state space criterion. *Wireless Personal Communications An International Journal*, *40*(2), 191–204.

Zha, D. (2007). Robust multiuser detection method based on least p-norm state space criterion. *Wireless Personal Communications*, *40*(2), 191–204.

Zhang, X. D. (2004). *Matrix analysis and applications*. (pp. 59–64).

Zhong, X., Premkumar, A. B., & Madhukumar, A. S. (2013). Particle filtering for acoustic source tracking in impulsive noise with alpha-stable process. *IEEE Sensors Journal*, *13*(2), 589–600.

Zimmermann, M., & Dostert, K. (2002). Analysis and modeling of impulsive noise in broad-band powerline communications. *IEEE Transactions on Electromagnetic Compatibility*, *44*(1), 249–258.