Neural Comput & Applic (2013) 22:569–576
DOI 10.1007/s00521-012-0873-x

Low complexity adaptive forgetting factor for online sequential extreme learning machine (OS-ELM) for application to nonstationary system estimations

Jun-seok Lim · Seokjin Lee · Hee-Suk Pang

Received: 4 August 2011 / Accepted: 27 January 2012 / Published online: 9 February 2012

Abstract Huang et al. (2004) has recently proposed an on-line sequential ELM (OS-ELM) that enables the extreme learning machine (ELM) to train data one-by-one as well as chunk-by-chunk. OS-ELM is based on recursive least squares-type algorithm that uses a constant forgetting factor. In OS-ELM, the parameters of the hidden nodes are randomly selected and the output weights are determined based on the sequentially arriving data. However, OS-ELM using a constant forgetting factor cannot provide satisfactory performance in time-varying or nonstationary environments. Therefore, we propose an algorithm for the OS-ELM with an adaptive forgetting factor that maintains good performance in time-varying or nonstationary environments. The proposed algorithm has the following advantages: (1) the proposed adaptive forgetting factor requires minimal additional complexity of O(N) where N is the number of hidden neurons, and (2) the proposed algorithm with the adaptive forgetting factor is comparable with the conventional OS-ELM with an optimal forgetting factor.

Keywords Extreme learning machine · OS-ELM · RLS adaptive forgetting factor

J. Lim (✉) · H.-S. Pang
Department of Electronics Engineering, Sejong University, 98 Kunja, Kwangjin, Seoul 143-747, Korea
e-mail: jslim@sejong.ac.kr

S. Lee
School of Electrical Engineering, Seoul National University, Seoul, Korea

1 Introduction

Researchers have extensively discussed single-hidden-layer feedforward neural networks (SLFNs) [1–5]. In the conventional SLFN learning theory, all the hidden-node parameters (the input weights $\mathbf{a}_i$ linking the input layer to the hidden layer and the biases b_i or the centers $\mathbf{a}_i$ and the impact factors b_i of the RBF hidden nodes) need to be tuned. Several researchers have found out that the input weights or centers $\mathbf{a}_i$ need not be tuned [6–9]. For example, Lowe [6] from an interpolation point of view found out that the centers $\mathbf{a}_i$ of RBF hidden nodes can be randomly selected from the training data instead of tuning. Igelnik and Pao [7] proposed a random vector version of functional-link (RVFL) net. Based on these previous works, Huang et al. [1] recently proposed a new learning algorithm named the extreme learning machine (ELM) in SLFN. In ELM, all the hidden-node parameters $\mathbf{a}_i$ and b_i are randomly generated independently of the target functions and the training patterns, while the above-mentioned learning methods [6–9] only randomly select the input weights or centers $\mathbf{a}_i$ [10]. In addition, compared to the popular back-propagation (BP) algorithm and support vector machine (SVM)/least square SVM (LS-SVM), ELM has several salient features [11]: (1) ease of use. No parameters need to be manually tuned except predefined network architecture; (2) faster learning speed. Most training can be completed in milliseconds, seconds, and minutes (for large-scale complex applications); (3) suitable for almost all nonlinear activation functions. Almost all piecewise continuous (including discontinuous, differential, nondifferential functions) can be used as activation functions in ELM; and (4) suitable for fully complex activation functions. Fully complex functions can also be used as activation functions in ELM.

The ELM was initially proposed to use batch learning (like many other SLFN applications); however, batch learning is usually a time consuming process that requires many iterations through the training data. In many real-world applications, learning has to be an ongoing process since a complete set of data is not readily available. Whenever new data arrives, the batch learning method needs to be retrained with prior and additional data (a process that requires significant time).

Lim developed a recursive-learning algorithm for a complex ELM [12]. Liang et al. [13] developed an on-line sequential learning version for the batch-type ELM [14–16] called the OS-ELM. In both Lim's algorithm and the OS-ELM, the input weights and the hidden-node biases are randomly generated; subsequently, output weights are then analytically determined. In both algorithms, all the hidden-node parameters are independent of the training data (as well as each other). Lim's algorithm can only sequentially learn the training data, that is, one-by-one. However, the OS-ELM can sequentially learn the training data one-by-one as well as chunk-by-chunk with a fixed or varying length. These two algorithms use the same recursive least squares method, although the OS-ELM uses more varied types of data updates.

The recursive least-square (RLS) algorithm is well known for its good convergence property and small mean square error (MSE) in stationary environments. However, the RLS using a constant forgetting factor (FF) cannot provide satisfactory performance in time-varying or non-stationary environments.

An RLS method with an adaptive FF needs to be developed in order to enhance the performance in non-stationary environments. A best example for doing this is described in [17]. This method shows good tracking performance. However, it suffers from heavy computational loads. In order to overcome the disadvantages, Song et al. [18] proposed the modified adaptive FF–RLS (MAFF–RLS) method. Lee et al. [19] proposed an improved MAFF–RLS method that uses the normalization technique found in the LMS method. The algorithm improves the dynamics of the adaptive FF. If we apply the adaptive FF to the OS-ELM, then we can expect the performance of the OS-ELM to improve in stationary and nonstationary environments. However, this still requires a heavy additional complexity, because the adaptive FF taken from [17–19] needs additional complexity. It is revealed to be more complex than the traditional RLS method.

In this study, we propose an algorithm that achieves good performance in both stationary and nonstationary situations with complexity comparable to that of RLS. In particular, the following contributions have been made in this paper.

1. An adaptive algorithm is proposed to calculate the time-variant FF for OS-ELM in order to achieve good performance in both stationary and nonstationary situations.
2. The proposed adaptive FF requires additional complexity of O(N) while the conventional adaptive FF requires additional complexity of O(N^2) where N is the number of hidden neurons.
3. The proposed algorithm with the adaptive FF is comparable with the conventional OS-ELM with an optimal FF.

2 Review of the online sequential ELM (OS-ELM)

Given a series of arbitrary training samples $(\mathbf{z}_i, d_i)$, $i = 1, 2, \ldots, N$, where $\mathbf{z}_i \in \mathrm{R}^p$ and $d_i \in \mathrm{R}^1$, the actual outputs of the SLFN using activation function $g_c(z)$ for these N training data is given by

$$\sum_{k=1}^{\tilde{N}} \beta_k g_c(\mathbf{w}_k \cdot \mathbf{z}_i + b_k) = d_i, \quad i = 1, \ldots, N, \tag{1}$$

where column vector $\mathbf{w}_k \in \mathrm{R}^p$ is the input weight vector connecting the input layer neurons to the kth hidden neuron, $\beta_k \in \mathbf{R}^1$ is the output weight vector connecting the kth hidden neuron and the output neuron, and $\mathrm{b}_k \in \mathrm{R}^1$ is the bias of the kth hidden neuron. $\mathbf{w}_k \cdot \mathbf{z}_i$ denotes the inner product of column vectors $\mathbf{w}_k$ and $\mathbf{z}_i$. The above N equations can be written compactly as:

$$\mathbf{H}\beta = \mathbf{d}, \tag{2}$$

where:

$$\mathbf{H}\left(\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}, \mathbf{z}_1, \ldots, \mathbf{z}_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}}\right) = \begin{bmatrix} g_c(\mathbf{w}_1 \cdot \mathbf{z}_1 + b_1) & \cdots & g_c(\mathbf{w}_{\tilde{N}} \cdot \mathbf{z}_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g_c(\mathbf{w}_1 \cdot \mathbf{z}_N + b_1) & \cdots & g_c(\mathbf{w}_{\tilde{N}} \cdot \mathbf{z}_N + b_{\tilde{N}}) \end{bmatrix}, \tag{3}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix} \quad \text{and} \quad \mathbf{d} = \begin{bmatrix} d_1^T \\ \vdots \\ d_N^T \end{bmatrix}. \tag{4}$$

Liang et al. [13] discuss online sequential adaptation strategies for the ELM equation in (2). First, they derive an algorithm using a RLS based on the matrix inversion lemma [17]. In [13], the one-by-one OS-ELM algorithm is summarized as:

1. Initial $\mathbf{P}(0), \hat{\beta}(0)$ and FF λ, where we can apply the initialization phase in [13].

2. For each observation ($\mathbf{z}(n)$, d(n)) and $n = 1, \ldots,$

   (a) Calculate the hidden layer output vector.

   (b) $\mathbf{h}(n) = [g_c(\mathbf{w}_1 \cdot \mathbf{z}_1 + b_1) \quad \cdots \quad g_c(\mathbf{w}_{\tilde{N}} \cdot \mathbf{z}_N + b_{\tilde{N}})]^T$.

   (c) Update output weight $\hat{\beta}(n)$ based on the RLS algorithm:

$$\mathbf{k}(n) = \frac{\mathbf{P}(n-1)\mathbf{h}(n)}{\lambda + \mathbf{h}^T(n)\mathbf{P}(n-1)\mathbf{h}(n)}. \tag{5}$$

$$e(n) = \mathrm{d}(n) - \mathbf{h}^T(n)\hat{\beta}(n). \tag{6}$$

$$\hat{\beta}(n) = \hat{\beta}(n-1) + \mathbf{k}(n)e(n). \tag{7}$$

$$\mathbf{P}(n) = \frac{1}{\lambda}\left[\mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{h}^T(n)\mathbf{P}(n-1)\right]. \tag{8}$$

   (d) $n = n + 1$.

The above one-by-one OS-ELM can then be extended to chunk-by-chunk [13].

3 Low complexity adaptive forgetting factor OS-ELM (LAFF–OS-ELM)

3.1 Conventional AFF–RLS method

In the conventional RLS method, the FF λ is a constant between 0 and 1. In general, the FF controls the effective data window length, which is proportional to $\frac{1}{1-\lambda}$ [20]. Therefore, the larger the FF, the longer the effective data window. Although a long data window produces good estimation results in stationary environments, the long data window fails in time-varying or nonstationary environments and shows the need to control the FF to the environments adaptively.

By using the adaptive FF recursive least squares (AFF−RLS) method, the FF is updated recursively in order to adjust itself to nonstationary circumstances. The goal of the AFF–RLS method is to calculate the FF that optimizes the mean square of the priori estimation error. The optimized cost function is determined by [18, 19]:

$$J'(n) = \frac{1}{2}E\left[|e(n)|^2\right], \tag{9}$$

where $E[\ ]$ is the expectation operator, and $e(k)$ is the priori estimation error, as stated above.

For optimization, we can take the partial derivative of the cost function with respect to the FF λ:

$$\nabla_\lambda(n) = \frac{\partial J'(n)}{\partial \lambda} = -\mathbf{\Psi}^T(n-1)\mathbf{h}(n)e(n), \tag{10}$$

where $\mathbf{\Psi}(n) = \frac{\partial \boldsymbol{\beta}(n)}{\partial \lambda}$ and $\mathbf{S}(n) = \frac{\partial \mathbf{P}(n)}{\partial \lambda}$. $\mathbf{\Psi}(n)$ and $\mathbf{S}(n)$ can be updated recursively by [17, 18]:

$$\mathbf{\Psi}(n) = \left[\mathbf{I} - \mathbf{k}(n)\mathbf{h}^T(n)\right]\mathbf{\Psi}(n-1) + \mathbf{S}(n)\mathbf{h}(n)e(n). \tag{11}$$

$$\mathbf{S}(n) = \lambda^{-1}(n-1)\left[\mathbf{I} - \mathbf{k}(n)\mathbf{h}^T(n)\right]\mathbf{S}(n-1)\left[\mathbf{I} - \mathbf{h}(n)\mathbf{k}^T(n)\right] + \lambda^{-1}(n-1)\mathbf{k}(n)\mathbf{k}^T(n) - \lambda^{-1}(n-1)\mathbf{P}(n). \tag{12}$$

The update process of the FF can be calculated by [18, 19]:

$$\lambda(n) = \lambda(n-1) - \alpha\mathbf{\Psi}^T(n-1)\mathbf{h}(n)e(n)\Big|_{\lambda_-}^{\lambda_+}, \tag{13}$$

where α is a small constant, λ_+ is the upper limit, and λ_- is the lower limit of the FF. The FF-RLS algorithm can be summarized by the following equations:

$$e(n) = \mathrm{d}(n) - \mathbf{h}^T(n)\hat{\beta}(n). \tag{14}$$

$$\mathbf{k}(n) = \frac{\mathbf{P}(n-1)\mathbf{h}(n)}{\lambda(n-1) + \mathbf{h}^T(n)\mathbf{P}(n-1)\mathbf{h}(n)}. \tag{15}$$

$$\hat{\beta}(n) = \hat{\beta}(n-1) + \mathbf{k}(n)e(n). \tag{16}$$

$$\mathbf{P}(n) = \lambda^{-1}(n-1)\left[\mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{h}^T(n)\mathbf{P}(n-1)\right]. \tag{17}$$

$$\mathbf{S}(n) = \lambda^{-1}(n-1)\left[\mathbf{I} - \mathbf{k}(n)\mathbf{h}^T(n)\right]\mathbf{S}(n-1)\left[\mathbf{I} - \mathbf{h}(n)\mathbf{k}^T(n)\right] + \lambda^{-1}(n-1)\mathbf{k}(n)\mathbf{k}^T(n) - \lambda^{-1}(n-1)\mathbf{P}(n) \tag{18}$$

$$\mathbf{\Psi}(n) = \left[\mathbf{I} - \mathbf{k}(n)\mathbf{h}^T(n)\right]\mathbf{\Psi}(n-1) + \mathbf{S}(n)\mathbf{h}(n)e(n). \tag{19}$$

$$\lambda(n) = \lambda(n-1) - \alpha\mathbf{\Psi}^T(n-1)\mathbf{h}(n)e(n)\Big|_{\lambda_-}^{\lambda_+}. \tag{20}$$

The AFF–RLS method can also be applied to learn the training data point-by-point or chunk-by-chunk because this algorithm uses the same recursive algorithm as the OS-ELM.

3.2 Effects of the forgetting factor (FF) in RLS and meaning of the adaptive FF

In the RLS problem, the normal equations [17] are written in matrix form as

$$\mathbf{R}(n)\hat{\beta}(n) = \mathbf{z}(n), \tag{21}$$

where $\mathbf{R}(n) = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{h}(i)\mathbf{h}^T(i)$ and $\mathbf{z}(n) = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{h}(i)d(i)$. As we mentioned before, the effective data window length can be derived from the FF as $l_w = \frac{1}{1-\lambda}$ [20]. By the effective data window length concept, it can be assumed that $\sum_{i=1}^{n} \lambda^{n-i}\mathbf{h}(i)\mathbf{h}^T(i) \cong \sum_{i=n-l_w+1}^{n} \lambda^{n-i}\mathbf{h}(i)\mathbf{h}^T(i)$ and $\sum_{i=1}^{n} \lambda^{n-i}\mathbf{h}(i)d(i) \cong \sum_{i=n-l_w+1}^{n} \lambda^{n-i}\mathbf{h}(i)d(i)$. It works like a moving average with a finite set of data. The window length becomes large with a small λ and vice versa [21, 22]. If any disturbance or variation in the signal or the

system happens, the estimation error $e(n)$ increases and λ is decreased by (13) and the effect of disturbed samples is decreased in calculation of $\mathbf{R}(n)$ and $\mathbf{z}(n)$. If data become stationary after the disturbance, the estimation error $e(n)$ decreases and λ is increased so that the effective window length goes longer. From the estimation variance point of view, the shorter data window length due to the small λ causes an increase of the estimation variance and vice versa [21, 22]. Therefore, the fixed FF with a single value cannot achieve both the low level of the estimation variance and the robustness against the disturbance or the variation in the signal or the system. Consequently, the adaptive FF can be a more appropriate solution in order to deal with these aspects.

Many papers [21–23] described the adaptive FF capability of both the low level of the estimation variance and the robustness against the disturbance or the variation in the signal or the system. The experiment results in this paper also show that OS-ELM with AFF can adapt itself to the variation or the disturbance in the signal or the system.

3.3 Reduced complexity AFF–RLS method

The conventional AFF–RLS method shows good performance; however, it needs to perform $9\tilde{N}^2 + 7\tilde{N}$ multiplications. This means that the introduction of the adaptive FF makes this method more complex than the conventional RLS that needs only $2.5\tilde{N}^2 + 3\tilde{N}$ multiplications.

In order to reduce this cost, an approximated version of the AFF–RLS was proposed in [18]. According to [18], $\mathbf{C}(n) = \mathbf{I} - \mathbf{k}(n)\mathbf{h}^T(n)$ can be approximated as simple diagonal matrix, $c(n)\mathbf{I}$, where $c(n) = 1 - \frac{1}{\tilde{N}}\mathbf{h}^T(n)\mathbf{k}(n)$. This approximation can reduce the number of multiplications due to the many $\mathbf{I} - \mathbf{k}(n)\mathbf{h}^T(n)$ calculations in the conventional AFF–RLS method. Using this approximation, the $\mathbf{S}(n), \mathbf{\Psi}(n)$, and $\lambda(n)$ can be calculated with a smaller complexity. We represent the approximated $\mathbf{S}(n), \mathbf{\Psi}(n)$ and $\lambda(n)$ as $\mathbf{S}(n), \mathbf{\Psi}(n)$, and $\bar{\lambda}(n)$, respectively. This reduced complexity method is called the modified AFF–RLS (MAFF–RLS) method.

3.4 Normalized adaptive forgetting factor (FF) method

Although the MAFF–RLS reduces the complexity; however, this approximation method suffers from an estimation error in $\nabla_\lambda(n)$. Therefore, the MAFF–RLS method has a severe 'gradient error amplification' problem. Lee et al. [19] proposed a method that resolves this problem by introducing normalization to the update equation found in (20). The update equation proposed by Lee et al. [19] is:

$$\bar{\lambda}(n) = \bar{\lambda}(n-1) + \frac{\alpha}{\left|\mathbf{\Psi}^T(n-1)\mathbf{h}(n)\right|^2} Re\left[\mathbf{\Psi}^T(n-1)\mathbf{h}(n)e(n)\right]\Big|_{\lambda_-}^{\lambda_+}. \tag{22}$$

Lee et al. [19] shows that the normalized adaptive FF works well when applied to the MAFF–RLS method. The proposed algorithm by Lee et al. is summarized as the modified normalized adaptive FF–RLS (MNAFF–RLS) by:

1. The first 4 steps are the same as (14)–(17)
2. Update the FF as follows:

$$c(n) = 1 - \frac{\mathbf{h}^T[n]\mathbf{k}[n]}{\tilde{N}}. \tag{23}$$

$$\boldsymbol{S}(n) = \left[\bar{\lambda}(n-1)\right]^{-1}|c(n)|^2\mathbf{S}(n-1) - \left[\bar{\lambda}(n-1)\right]^{-1}c(n)\mathbf{P}(n). \tag{24}$$

$$\mathbf{\Psi}(n) = c(n)\mathbf{\Psi}(n-1) + \mathbf{S}(n)\mathbf{h}(n)e(n). \tag{25}$$

$$\bar{\lambda}(n) = \bar{\lambda}(n-1) + \frac{\alpha}{\left|\mathbf{\Psi}^T(n-1)\mathbf{h}(n)\right|^2} Re\left[\mathbf{\Psi}^T(n-1)\mathbf{h}(n)e(n)\right]\Big|_{\lambda_-}^{\lambda_+}. \tag{26}$$

For chunk-by-chunk update, we can replace $c(n) = 1 - \frac{\mathbf{h}^T(n)\mathbf{k}(n)}{\tilde{N}}$ with $c(n) = 1 - \frac{1}{\tilde{N}(t-i+1)}\sum_i^t \mathbf{h}_i^T(n)\mathbf{k}_i(n)$, where $\mathbf{h}_i(n)$ is the$i$th column of the chunk data matrix $\mathbf{H}(n)_{\tilde{N}\times t}$ and $\mathbf{k}_i(n)$ is the ith column of the Kalman gain matrix $\mathbf{K}(n)_{\tilde{N}\times t} = \mathbf{P}(n)_{\tilde{N}\times\tilde{N}}\mathbf{H}(n)_{\tilde{N}\times t}$.

3.5 Simplified modified normalized adaptive forgetting factor–RLS (SMNAFF–RLS) method

We can further reduce the complexity in the FF adjustment process for the MNAFF–RLS with a proper assumption and a property of the RLS algorithm. If the sampling interval is sufficiently small, we can use the approximation shown in (27) with little error [24].

$$\mathbf{S}(n-1)\mathbf{h}(n) \cong \mathbf{S}(n-1)\mathbf{h}(n-1). \tag{27}$$

The relation in (27) is valid within the short time distance between $\mathbf{h}(n)$ and $\mathbf{h}(n-1)$; however, the time distance between $\mathbf{h}(n)$ and $\mathbf{h}(n-1)$ may be larger depending on the coherence distance of the system $\mathbf{h}(n)$. The simulation section shows the effect of the time interval between $\mathbf{h}(n)$ and $\mathbf{h}(n-1)$.

When we set $\hat{\underline{\mathbf{q}}}(n) = \mathbf{S}(n)\mathbf{h}(n)$ and use the relationship for $\mathbf{k}(n) = \mathbf{P}(n)\mathbf{h}(n)$ from [17], we derive a new equation, shown in (30), by multiplying both sides of (24) by $\mathbf{h}(n)$. Therefore, we can use (30) instead of (24).

$$\begin{aligned}\mathbf{S}(n)\mathbf{h}(n) = &\left[\bar{\lambda}(n-1)\right]^{-1}|c(n)|^2\mathbf{S}(n-1)\mathbf{h}(n)\\ &-\left[\bar{\lambda}(n-1)\right]^{-1}c(n)\mathbf{P}(n)\mathbf{h}(n).\end{aligned} \tag{28}$$

$$\begin{aligned}\mathbf{S}(n)\mathbf{h}(n) \cong &\left[\bar{\lambda}(n-1)\right]^{-1}|c(n)|^2\mathbf{S}(n-1)\mathbf{h}(n-1)\\ &-\left[\bar{\lambda}(n-1)\right]^{-1}c(n)\mathbf{P}(n)\mathbf{h}(n).\end{aligned} \tag{29}$$

$$\hat{\underline{\mathbf{q}}}(n) = \left[\bar{\lambda}(n-1)\right]^{-1}\left[|c(n)|^2\hat{\underline{\mathbf{q}}}(n-1) - c(n)\mathbf{k}(n)\right]. \tag{30}$$

The proposed simplification shown in (27) results in a new simplified adaptive FF RLS, called the simplified modified normalized adaptive FF–RLS (SMNAFF–RLS) method summarized as follows:

1. The first 4 steps are the same as (14)–(17)
2. Update the FF as follows:

$$c(n) = 1 - \frac{\mathbf{h}^T[n]\mathbf{k}[n]}{\tilde{N}}. \tag{31}$$

$$\hat{\underline{\mathbf{q}}}(n) = \left[\bar{\lambda}(n-1)\right]^{-1}\left[|c(n)|^2\hat{\underline{\mathbf{q}}}(n-1) - c(n)\mathbf{k}(n)\right]. \tag{32}$$

$$\bar{\Psi}(n) = c(n)\bar{\Psi}(n-1) + \hat{\underline{\mathbf{q}}}(n)e(n) \tag{33}$$

$$\begin{aligned}\bar{\lambda}(n) = &\bar{\lambda}(n-1) + \frac{\alpha}{\left|\mathbf{\Psi}^T(n-1)\mathbf{h}(n)\right|^2}\\ &Re\left[\mathbf{\Psi}^T(n-1)\mathbf{h}(n)e(n)\right]\Big|_{\lambda_-}^{\lambda_+}.\end{aligned} \tag{34}$$

In Table 1, we compare the complexity of the conventional RLS, the AFF–RLS [17], the MAFF–RLS [18], and the proposed SMNAFF–RLS. This table also compares the data memory space in word units [25]. The complexity of the MNAFF–RLS remains almost at the level of the MAFF–RLS itself [18, 19]. Table 1 shows that the proposed SMNAFF–RLS has the lowest complexity (of the four algorithms) and requires the least data space of the four algorithms.

Table 1 shows that the proposed SMNAFF–RLS algorithm is very useful in order for the conventional OS-ELM to estimate nonstationary systems. When the proposed SMNAFF–RLS is applied to the conventional OS-ELM, we call it SMNAFF–OS-ELM.

4 Simulation results

In this section, we show the performance of the proposed algorithm. In Sect. 4.1, we show the performance comparison. The comparison shows that the performance of the proposed simplified algorithm is almost same as that of the other adaptive forgetting factored algorithms and also shows that the approximation used in the proposed algorithm does not degrade the performance. In addition, we can confirm that the propose algorithm is comparable with the conventional OS-ELM with an optimal FF. In Sect. 4.2, we give an application example for the time-varying nonlinear system identification. From this example, it is found that the proposed algorithm can handle the variation or the disturbance effectively by the adaptive FF.

4.1 The identification of random walk channel

In this simulation, we compare the estimation error from the conventional OS-ELM with the estimation errors from the proposed SMNAFF–RLS-based ELM, AFF–RLS-based ELM, MAFF–RLS-based ELM, and MNAFF–RLS-based ELM. For this comparison, we set the optimal FF for the conventional OS-ELM. All the inputs and outputs have been normalized into the range $[-1, 1]$ [13, 26]. We use 50 hidden neurons and the sigmoid function as an activation function of $1/(1 + \exp(-(\mathbf{a}^T\mathbf{x} + b)))$. The input weights and biases are randomly chosen from the range $[-1, 1]$ [13, 26]. In order to average the performance, 100 independent autoregressive (AR) channels are generated. Here, we chose the AR channel model, since the optimal FF is known in [18]. Consider three-tap, equal power channels that are generated from the following AR model:

$$\mathbf{h}(n) = \sqrt{a}\mathbf{h}(n-1) + \mathbf{v}(n), \tag{35}$$

where $\sqrt{a} = 0.999$, and $\mathbf{v}[n]$ is zero mean circular complex white Gaussian noise with $E[\mathbf{v}\mathbf{v}^H] = (1 - a)$ [27]. With such a coefficient, the coherence time becomes approximately

Table 1 Complexity comparisons

	Total complex multiplication	Additionally required multiplication for FF update	Total data memory usage[a] (in units of B-bit words)	Additionally required data memory usage
RLS	$2.5\tilde{N}^2 + 3\tilde{N}$	0	$2\tilde{N}^2 + 4\tilde{N} + 5$	0
AFF–RLS	$9\tilde{N}^2 + 7\tilde{N}$	$6.5\tilde{N}^2 + 4\tilde{N}$	$7\tilde{N}^2 + 6\tilde{N} + 6$	$5\tilde{N}^2 + 2\tilde{N} + 1$
MAFF–RLS	$4\tilde{N}^2 + 5\tilde{N}$	$1.5\tilde{N}^2 + 2\tilde{N}$	$4\tilde{N}^2 + 6\tilde{N} + 8$	$2\tilde{N}^2 + 2\tilde{N} + 3$
SMNAFF–RLS	$2.5\tilde{N}^2 + 7\tilde{N}$	$4\tilde{N}$	$2\tilde{N}^2 + 6\tilde{N} + 8$	$2\tilde{N} + 3$

[a] Sample-based processing is assumed

20 sample steps [28]. Then, the minimum mean-squared error (MSE) corresponding to the conventional OS-ELM is achieved at $\lambda_{\text{opt}} \approx 1 - \sqrt{1-a}/\sigma_n\sigma_u$ [18], where σ_n is the SD of the additive noise in the system output and σ_u is the standard deviation of the system input.

Figure 1 illustrates the MSE results for the conventional OS-ELM, the AFF–RLS-based ELM [17], the MAFF–RLS-based ELM [18], the MNAFF–RLS-based ELM [19], and the proposed SMNAFF–RLS-based ELM, where the optimal FF for the conventional OS-ELM algorithm is obtained from [18]. The performance curve of the proposed SMNAFF–RLS remains very close to the conventional OS-ELM algorithm with the optimal FF as well as those of the AFF–RLS-based ELM [17] , the MAFF–RLS-based ELM [18], and the MNAFF–RLS-based ELM [19].

To show the effect of the estimation update interval in (27), we vary the update interval 1, 5, 10, 50, 100, and 500 samples, respectively. Figure 2 shows the results from these variations. This figure shows that the proposed method keeps the same performance as the MNAFF–RLS method in [19] to the coherent time of the channel $\mathbf{h}(n)$. From 20 to 100 samples, the MSE of the proposed algorithm becomes inferior to the MNAFF–RLS algorithm; however, the MSEs of two algorithms converge to the same level as the update interval becomes farther. In conclusion, the approximation in (27) is valid for the large sample interval as well as for the short sample interval.

4.2 The identification of a wiener system with an abrupt channel change

The Wiener system is a well-known and simple nonlinear system that consists of a series connection of a linear filter and a memoryless nonlinearity (see Fig. 3). Such a nonlinear channel can be encountered in digital satellite communications and in digital magnetic recording.

Traditionally, many neural network structures such as MLPs [10], recurrent neural networks [29], or piecewise linear networks [30] have tackled the nonlinear system identification problem. We consider a supervised identification problem, in which at a given instant, the linear channel coefficients change abruptly, in order to compare the tracking capabilities of the algorithms: During the first part of the simulation, the linear channel is $H_1(z) = -0.3 - 0.9z^{-1} + 0.8\ z^{-2} - 0.7\ z^{-3} - 0.6\ z^{-4}$; after receiving 500 symbols, it is changed into $H_2(z) = 0.6 - 0.7\ z^{-1} + 0.8\ z^{-2} - 0.9\ z^{-3} - 0.3\ z^{-4}$. Consequently, we simulate a nonstationary nonlinear channel environment that has a changing sequence of stationary → abrupt change → stationary. A binary signal ($\mathbf{x} \in \{-1 + 1\}$) is sent through this channel after which the signal is transformed nonlinearly according to the nonlinear function $y = \tanh(v)$, where v is the linear channel output. The Wiener system is treated as a black box from which only the input and output are known.

The system identification was performed by the proposed adaptive FF OS-ELM as well as by OS-ELMs with a fixed FF of 1, 0.975, and 0.95, respectively. Both methods have 50 hidden neurons and use the sigmoid function as an activation function of $1/(1 + \exp(-(\mathbf{a}^T\mathbf{x} + b)))$. For both methods, the input weights and biases are randomly chosen from the range [−1, 1]. In order to average the performance, we ran 100 trials.

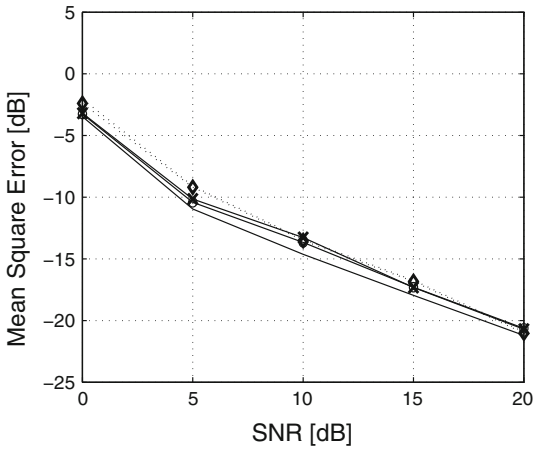


Fig. 1 MSE comparison for random walk channel estimation (*solid line* the conventional RLS with the optimal FF, *dash line* AFF–RLS, *dash* and *diamond line* MAFF–RLS, *open circle* MNAFF–RLS, *times symbol* the proposed algorithm)

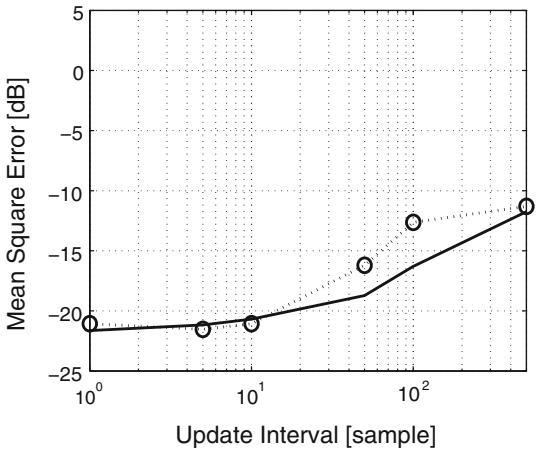


Fig. 2 MSE comparison for update interval variations (*solid line* MNAFF–RLS and *dash* and *circle* the proposed algorithm)

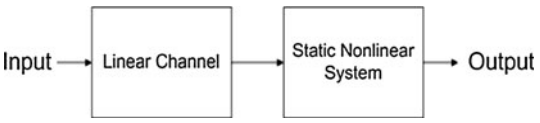
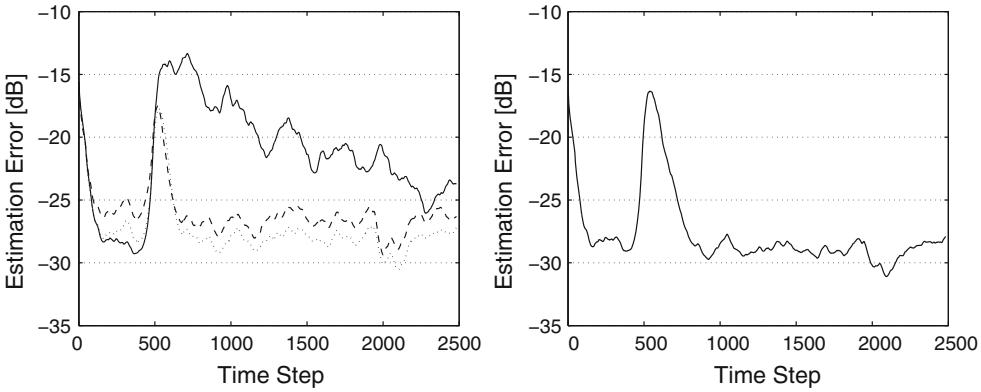


Fig. 3 Wiener channel model



(a) OS-ELM with fixed forgetting factors **(b)** ProposedOS-ELM

Fig. 4 Time-varying nonlinear channel estimation comparison between the proposed OS-ELM and the OS-ELM with fixed FFs (*solid* FF = 1, *dotted* FF = 0.975, *dashed* FF = 0.95)

Figure 4a shows the results from the OS-ELMs. Figure 4b shows the result from the proposed adaptive forgetting factored OS-ELM. From Fig. 4a, we can see that the smaller FF keeps better track when the abrupt change happens. However, the smaller fixed FF makes the estimation error worse in the stationary interval. On the contrary, the proposed algorithm tracks the changing system well regardless of the abrupt disturbance. Additionally, it maintains good estimation error. From the result, we can confirm that the proposed algorithm adapts the FF depending on the circumstances.

5 Conclusions

The OS-ELM algorithm has been proposed for training data recursively in on-line. However, the OS-ELM algorithm that uses a constant FF cannot provide satisfactory performance in time-varying or nonstationary environments. In this paper, we apply a new adaptive FF to the OS-ELM and propose a new algorithm that maintains good performance in time-varying or nonstationary environments with little additional complexity.

The proposed algorithm has the following advantages: (1) the proposed adaptive FF requires additional complexity of O(N) while the conventional adaptive FF requires additional complexity of O(N^2) where N is the number of hidden neurons; and (2) the proposed algorithm with the adaptive FF is comparable with the conventional OS-ELM with an optimal FF.

Finally, in the future, we aim to investigate the tracking performance in an audio application with the time-varying room impulse response, as well as in a nonlinear array processing with the time-varying interference.

Acknowledgments This work was supported by the National Research Foundation of Korea(NRF) grant funded.

References

1. Huang G-B, Zhu Q-Y, Siew C-K (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of the international joint conference on neural networks, Budapest, pp 25–29
2. Park J, Sandberg IW (1991) Universal approximation using radial-basis-function networks. Neural Comput 3:246–257
3. Barron AR (1993) Universal approximation bounds for superpositions of a sigmoid function. IEEE Trans Inf Theory 39: 930–945
4. Leshno M, Lin VY, Pinkus A, Schocken S (1993) Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Netw 6:861–867
5. Huang G-B, Wang DH, Lan Y (2011) Extreme learning machines: a survey. Int J Mach Lean Cybern 2:107–122
6. Lowe D (1989) Adaptive radial basis function nonlinearities and the problem of generalisation. In: Proceedings of the first IEE international conference on artificial neural networks, London, pp 171–175
7. Igelnik B, Pao Y-H (1995) Stochastic choice of basis functions in adaptive function approximation and the functional-link net. IEEE Trans Neural Netw 6:1320–1329
8. Baum E (1988) On the capabilities of multilayer perceptrons. J Complex 4:193–215
9. Ferrari S, Stengel RF (2005) Smooth function approximation using neural networks. IEEE Trans Neural Netw 16:24–38
10. Huang G-B, Li M-B, Chen L, Siew C-K (2008) Incremental extreme learning machine with fully complex hidden nodes. Neurocomputing 71:576–583
11. ELM web portal. http://www.ntu.edu.sg/home/egbhuang
12. Lim J, Jeon J, Lee S (2006) Recursive complex extreme learning machine with widely linear processing for nonlinear channel equalizer. LNCS 3973:128–134
13. Liang N-Y, Huang G-B, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw 17:1411–1423
14. Huang G-B, Zhu Q-Y, Mao KZ, Siew C-K, Saratchandran P, Sundararajan N (2006) Can threshold networks be trained directly? IEEE Trans Circuits Syst II Exp Briefs 53:187–191
15. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70:1–3
16. Huang G-B, Chen L, Siew C-K (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17:879–892
17. Haykin S (2002) Adaptive filter theory, 4th edn. Prentice Hall, NJ

18. Song S, Sung KM (2007) Reduced complexity self-tuning adaptive algorithms in application to channel estimation. IEEE Trans Commun 55:1448–1452
19. Lee S, Lim J, Sung K-M (2009) A low-complexity AFF-RLS algorithm using a normalization technique. IEICE Electron Exp 6:1774–1780
20. Niedzwiecki M (2000) Idenrification of time-varying process. Wiley, West Sussex
21. Paleologu C, Benesty J, Ciochina S (2008) A robust variable forgetting factor recursive least-squares algorithm for system identification. IEEE Signal Process Lett 15:597–600
22. Tuan P, Lee S, Hou W (1997) An efficient on-line thermal input estimation method using Kalman filter and recursive least square algorithm. Inverse Probl Eng 5:309–333
23. Kim H-S, Lim J-S, Baek S, Sung K-M (2001) Robust Kalman filtering with variable forgetting factor against impulsive noise. IEICE Trans Fundam E84-A:363–366
24. Yang B (1995) Projection approximation subspace tracking. IEEE Trans Signal Process 43:95–107
25. Lee K, Gan W, Kuo S (2009) Subband adaptive filtering theory and implementation. Wiley, West Sussex
26. Huang G-B, Chen L (2007) Convex incremental extreme learning machine. Neurocomputing 70:3056–3062
27. Han K, Lee S, Lim J, Sung K (2004) Channel estimation for OFDM with fast fading channels by modified Kalman filter. IEEE Trans Consumer Electron 50:443–449
28. Rappaport T (1996) Wireless communications principles and practice. Prentice Hall, NJ
29. Adali T, Liu X (1997) Canonical piecewise linear network for nonlinear filtering and its application to blind equalization. Signal Process 61:145–155
30. Holland PW, Welch RE (1997) Robust regression using iterative reweighted least squares. Commun Stat Theory Methods A 6:813–827