Neurocomputing 000 (2017) 1–9

Contents lists available at ScienceDirect

Neurocomputing

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

The selection of input weights of extreme learning machine: A sample structure preserving point of view

Wenhui Wang[a,*], Xueyi Liu[b]

[a] Basic Department, Zhejiang University of Water Resources and Electric Power, Hangzhou 310018, China
[b] Department of Mathematics, China Jiliang University, Hangzhou 310018, China

ARTICLE INFO

Article history:
Received 30 September 2015
Revised 11 June 2016
Accepted 16 June 2016
Available online xxx

Keywords:
Monte Carlo sampling
Quasi-Monte Carlo sequence
Extreme learning machine
Distance preserving
Generalization performance
Orthogonal projection

ABSTRACT

The random assignment strategy for input weights has brought extreme learning machine (ELM) many advantages such as fast learning speed, minimal manual intervention and so on. However, the Monte Carlo (MC) based random sampling method that is typically used to generate input weights of ELM has poor capability of sample structure preserving (SSP), which will degenerate the learning and generalization performance. For this reason, the Quasi-Monte Carlo (QMC) method is revisited and it is shown that the distortion error of QMC projection can obtain faster convergence rate than that of MC for relatively low-dimensional problems. Further, a unified random orthogonal (RO) projection method is proposed, and it is shown that such RO method can always provide the optimal transformation in terms of minimizing the loss of all the distances between samples. Experimental results on real-world benchmark data sets verify the rationality of theoretical analysis and indicate that by enhancing the SSP capability of input weights, QMC and RO projection methods tend to bring ELM algorithms better generalization performance.

1. Introduction

During the past years, feedforward neural networks (FNNs) have been extensively studied and widely used in various areas of machine learning. FNN theories show that single hidden layer feedforward neural networks (SLFNs) are sufficient to approximate any function with any desired accuracy providing some associated conditions being satisfied [1,2]. Back propagation (BP) algorithm [3] is probably the most representative training algorithm, which use the gradient descent method to update the weights in SLFNs. However, such gradient-based iterative learning algorithm is generally far slower than required, which has become a major drawback hampering its more extensive applications especially for large-scale problems.

Extreme learning machine (ELM), proposed by Huang et al. [4,5], is an alternative learning algorithm for SLFNs, which can overcome the learning issues faced by classical learning methods. ELM can be considered as an implementation of the well-known idea of mapping an original input space into a feature space by a nonlinear map, in which the tackled problem can be solved more easily. Theoretical analysis [6] shows that the parameters of the nonlinear map (input weights and biases of hidden neurons) should be fine-tuned to assure SLFNs the universal approximators. Unlike traditional learning theories, Huang et al. [4,7] proved that with the nonlinear map being randomly initialized and unchanged during the whole training process, SLFNs have universal approximation capability. The weights of the output layer are obtained by solving a linear system based on applying a Moore–Penrose's generalized inverse [8]. Therefore, the computational cost is much lower than when using other traditional learning algorithms. Nowadays such SLFNs with random hidden neurons provide us an exciting way to tackling the challenges of big data. And Akusok et al. [9] have recently presented a complete toolbox for big data application for such randomization based ELM solution, which are applicable to a wide range of machine learning problems and hence provide a solid ground for tackling numerous big data challenges.

The randomness of the hidden neurons is one of the key points that makes ELM stand out from other classical algorithms, especially for its high efficiency and good generalization. ELM theories argue that "random hidden neurons" capture the essence of some brain learning mechanisms as well as the intuitive sense that the efficiency of brain learning need not rely on computing power of neurons [10]. Liu et al. [11] and Lin et al. [12] assessed the feasibility of such random-node-based learning method theoretically. Wang et al. [13] have verified the positive effect of the random in-

* Corresponding author.
E-mail address: zjuwangwh@163.com (W. Wang).

http://dx.doi.org/10.1016/j.neucom.2016.06.079
0925-2312/

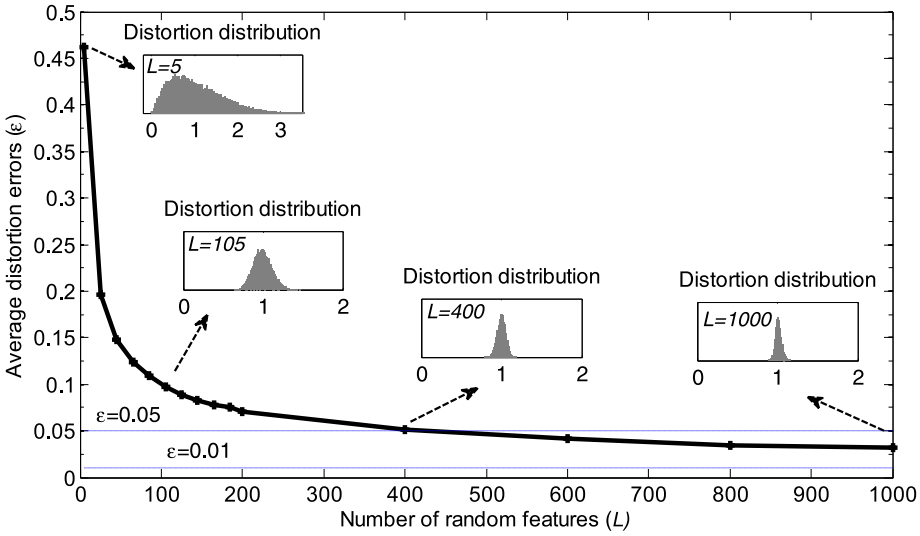

Fig. 1. The mean of distortion errors of random projection with different number of objective dimension (L) over 50 runs on Computer Activity data set; and the subplots herein depict the corresponding distortion distribution for one try.

put weights experimentally. On the other hand, the randomness of hidden nodes poses some new issues to be addressed in real applications, such as the oscillation of its performance [14], the effectiveness [15], the tendentiousness of selecting more hidden nodes than BP algorithm [16,17], and so on.

In this paper, we focus on another important property of random input weights in ELM: the distance information preserving capability. In the real implementation of ELM algorithm, the input weights are generally generated by pseudo-random number generators corresponding to a specific continuous distribution, which is also known as Monte Carlo (MC) sampling method. The well-known Johnson–Lindenstrauss (JL) lemma [18] gives an intuition explanation of the rationality of such MC-based random projection, which states that a set of N points in Euclidean space can be mapped both randomly and linearly to a space of $O(log(N)\varepsilon^{-2})$ dimension such that the distances between the points are distorted by at most $1+\varepsilon$, where $0 < \varepsilon < 1/2$.

Nevertheless, it should also be noted that when ε is relatively small, for instance, $10^{-1} - 10^{-2}$, the objective dimension would amount to a much larger number $(10^2 - 10^4)$ than expected in a good many real applications. To illustrate this point more explicitly, Fig. 1 gives the experimental results of how the average distortion error of all the pair-wise distances changes with respect to the number of random features (L) on Computer Activity data set coming from the Delve webpage http://www.cs.toronto.edu/~delve, which has 12 attributes and 8192 samples. It can be seen that the distortions are highly dispersed when the number of L is relatively small. As L increases, the distortions assemble together and the distortion errors converge to 0 gradually. Specially, the subplot corresponding to $L = 1000$ shows that the sample structure information could be well preserved in the 1000-dimensional random space. However, in practice, such large network size often leads to the over-fitting phenomena. For example, the carefully designed ELM model for Computer Activity case in previous work [4] owns only 125 hidden nodes. Obviously, it is of great importance to decrease the distortion error while not increasing the network size. Liu et al. [19] have also pointed out that the ELM algorithm can generalize as well as or better than the SVM algorithm for large sample cases, but it tends to oscillate harder and generalize worse than the SVM algorithm for small network size or small sample cases. The reason may lie in the poor sample structure preserving (SSP) capability of the random input weights of the ELM models with small network size.

In addition, it should be noted that a kind of improved versions of ELM have recently been developed based on manifold learning methods, such as manifold regularization [20,21], graph Laplacian [22,23] and Riemannian metric [24]. The common idea behind these works is to preserve the local neighbor structure information during learning the output weights of ELM. However, these efforts did not take into account the local structure loss caused by random input weights, which might be more essential for the final learning performance.

Hence, all the analysis above motivates us to investigate the SSP capability of such random projection, and further to propose new projection methods possessing better SSP capability to enhance the learning performance of ELM. The main contributions of this paper are: (1) to offer a new perspective to evaluate the quality of input weights of ELM; (2) to prove that the Quasi-Monte Carlo (QMC) method can provide better SSP capability than MC method for relatively low-dimensional cases; (3) to propose a new unified random orthogonal (RO) projection method for ELM that can always obtain the best SSP capability; and (4) to show through simulations that it is an effective way of enhancing the learning and generalization performance of ELM to improve the SSP performance of input weights. To the best of our knowledge, there is no systematic analysis of the SSP capability of the ELM random projection and its effects on the learning performance.

The rest of the paper is organized as follows. In Section 2, ELM is briefly introduced and the SSP capability of MC projection is discussed, while two new projection methods and their SSP capability are provided in Section 3. Section 4 presents the simulation studies on the proposed methods for regression and classification benchmark data sets. Some discussion about the existing related works is provided in Section 5 and finally the conclusions are drawn in Section 6.

2. ELM random features and the SSP capability

2.1. ELM random features

A SLFN with L hidden nodes (as shown in Fig. 2) can be represented as the following equations:

$$f(\mathbf{x}) = \sum_{i=1}^{L} \beta_i G(\mathbf{x}; \mathbf{w}_i, b_i) \quad (1)$$

Please cite this article as: W. Wang, X. Liu, The selection of input weights of extreme learning machine: A sample structure preserving point of view, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.06.079

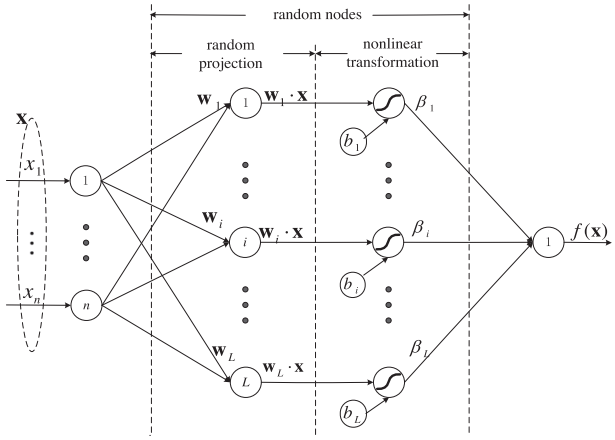

Fig. 2. Single-hidden layer feedforward network with additive random hidden nodes.

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{w}_i = (w_{i1}, w_{i1}, \ldots, w_{in})$; and the nonlinear feature function $G : \mathbb{R}^n \times \Omega \to \mathbb{R}$, parameterized by $(\mathbf{w}_i,\ b_i) \in \Omega$, is weighted by $\beta_i \in \mathbb{R}$. And $G(\mathbf{x};\ \mathbf{w}_i,\ b_i)$ calculates the output of the hidden node, and generally takes the additive form $G(\mathbf{x}; \mathbf{w}_i, b_i) = g(\mathbf{w}_i \cdot \mathbf{x} + b_i)$, where $g : \mathbb{R} \to \mathbb{R}$ is activation function.

Usually, parameters $(\mathbf{w}_i,\ b_i)$ and β_i in expression (1) can be trained using different popular methods such as back propagation [3], convex optimization (such as support vector machines [25]), and greedy algorithm (such as Adaboost [26], or matching pursuit [27]). Nevertheless, the computational complexity of all these methods still remains superlinear in the training sample size and cannot adapt to the steady growth of volumes of practical data sets.

In [7], it is shown that the SLFNs with input weights of the hidden nodes initialized by sampling from a continuous distribution and output weights calculated by solving a least square problem still maintain the universal capability of SLFNs. The input weights are randomly assigned and need not be tuned during the whole learning process.

Such hidden nodes and the corresponding mapping are called ELM random features. As shown in Fig. 2, ELM random features can be divided into two stages:

(i) the random projection stage

$$\mathbf{z} = \mathbf{W}\mathbf{x}$$

(ii) and the nonlinear transformation stage

$$\mathbf{h} = g(\mathbf{z} + \mathbf{b})$$

This paper is mainly focused on the SSP performance of the first stage: random projection stage. As pointed out previously, the random projection sequence $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L\}$ is usually obtained based on a pseudo-random MC method, so we call such random projection as MC projection and denote the ELM model with MC projection as ELM-MC in this paper.

2.2. SSP capability of MC projection

In ELM, uniform distribution $U[-1, 1]^n$ is the most often used one when generating the MC projection sequences. Theorem 1 gives the corresponding SSP capability under this distribution, which can be easily proved by following the proof method of JL Lemma. Here we will include the proof just for completeness.

Theorem 1. *(Convergence rate of MC projection) Suppose that $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L$ are random variables uniformly distributed over $[-1, 1]^n$, then for any $\mathbf{u} \in \mathbb{R}^n$, with probability at least $1 - \delta$,*

$$(1-\varepsilon)\frac{L\|\mathbf{u}\|^2}{3} \le \|\mathbf{W}\mathbf{u}\|^2 \le (1+\varepsilon)\frac{L\|\mathbf{u}\|^2}{3} \tag{2}$$

where $\mathbf{W} = [\mathbf{w}_1^T, \mathbf{w}_2^T, \ldots, \mathbf{w}_L^T]^T$ and $\varepsilon = \frac{2}{\sqrt{5\delta L}}$.

Proof. Suppose a random vector $\mathbf{w} = [w_1, w_2, \ldots, w_n] \sim U[-1, 1]^n$, then

$$\mathbb{E}[(\mathbf{w}\cdot\mathbf{u})^2] = \mathbb{E}\left[\left(\sum\nolimits_{i=1}^n w_i u_i\right)^2\right] = \sum\nolimits_{i=1}^n \mathbb{E}[w_i^2]u_i^2 = \|\mathbf{u}\|^2/3. \tag{3}$$

And

$$\mathbb{E}[(\mathbf{w}\cdot\mathbf{u})^4] = \mathbb{E}\left[\left(\sum_{i=1}^n w_i u_i\right)^4\right] = \sum_{i_1,i_2,i_3,i_4=1}^n \mathbb{E}[w_{i_1}w_{i_2}w_{i_3}w_{i_4}]u_{i_1}u_{i_2}u_{i_3}u_{i_4}$$
$$= \sum_{i=1}^n \mathbb{E}[w_i^4]u_i^4 + \sum_{i_1,i_2=1;i_1\neq i_2}^n \mathbb{E}[w_{i_1}^2 w_{i_2}^2]u_{i_1}^2 u_{i_2}^2$$
$$= \frac{1}{5}\sum_{i=1}^n u_i^4 + \frac{1}{9}\sum_{i_1,i_2=1;i_1\neq i_2}^n u_{i_1}^2 u_{i_2}^2$$
$$\le \frac{1}{5}\|\mathbf{u}\|^4$$

Hence,

$$\mathbb{D}[(\mathbf{w}\cdot\mathbf{u})^2] = \mathbb{E}\left[(\mathbf{w}\cdot\mathbf{u})^4\right] - \left(\mathbb{E}\left[(\mathbf{w}\cdot\mathbf{u})^2\right]\right)^2 \le \frac{1}{5}\|\mathbf{u}\|^4 - \frac{1}{9}\|\mathbf{u}\|^4$$
$$= \frac{4}{45}\|\mathbf{u}\|^4 \tag{4}$$

Combining Chebyshev inequality and Eqs. (3) and (4), we can get that, for any $\varepsilon > 0$,

$$P\left(\left|\frac{1}{L}\sum_{i=1}^L (\mathbf{w}_i\cdot\mathbf{u})^2 - \frac{\|\mathbf{u}\|^2}{3}\right| \ge \frac{\|\mathbf{u}\|^2}{3}\varepsilon\right) \le \frac{\mathbb{D}\left[\frac{1}{L}\sum_{i=1}^L (\mathbf{w}_i\cdot\mathbf{u})^2\right]}{\varepsilon^2\|\mathbf{u}\|^4/9}$$
$$= \frac{4}{5L\varepsilon^2}. \tag{5}$$

Let $\delta = 4/5L\varepsilon^2$, we can obtain that

$$P\left((1-\varepsilon)\frac{L\|\mathbf{u}\|^2}{3} \le \|\mathbf{W}\mathbf{u}\|^2 \le (1+\varepsilon)\frac{L\|\mathbf{u}\|^2}{3}\right) \ge 1-\delta$$

with $\varepsilon = 2/\sqrt{5\delta L}$. □

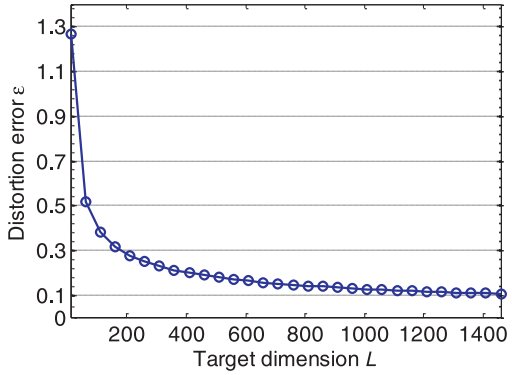

Fig. 3. The theoretical relationship between distortion error ε and the target dimension L with $\delta = 0.05$.

Theorem 1 shows that after the MC projection of ELM, the samples will averagely expand by $L/3$ times in sense of "$\|^*\|^2$", with the distortion error ε converges to 0 at the rate of $O(1/\sqrt{\delta L})$. Fig. 3 gives an illustration of the theoretical relationship between ε and L, given $\delta = 0.05$. It can be seen that to obtain small distortion error, i.e. good SSP capability of MC projection, ELM models with

large network size should be used, which will often be much larger than desired. Hence it is necessary to find new projection methods that can guarantee small distortion error with smaller network size.

3. Two alternative projection methods and their SSP capability

In this section, we will first revisit the Quasi-Monte Carlo (QMC) method and show its SSP capability, and further propose a unified random orthogonal (RO) projection method, which can provide the optimal SSP capability.

3.1. QMC projection

Here we will first give a brief introduction to the QMC method before discuss its SSP capability. The QMC method is a deterministic version of MC method. The aim of QMC method is to improve the convergence rate by using a deterministic low-discrepancy sequence. In this paper, we propose to use the low-discrepancy properties of QMC sequences to reduce the distortion error of the ELM random mapping.

For lack of space, only the necessary background of QMC method for understanding subsequent sections will be discussed. The interested readers are referred to the excellent reviews [28,29] for details.

Consider the problem of estimation of an integral

$$I(f) = \int_{[0,1]^n} f(\mathbf{x})d\mathbf{x}. \tag{6}$$

Note that if $\mathbf{x}$ is a random variable that follows $U[0, 1]^n$, then $I(f) = \mathbb{E}[f(\mathbf{x})]$. Hence, according to the law of large numbers, the approximation of the above integral can be given by

$$\hat{I}(f) = \frac{1}{L}\sum_{i=1}^{L} f(\mathbf{w}_i), \tag{7}$$

with $P = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L\}$ sampled independently from the $U[0, 1]^n$ distribution. This is the Monte Carlo Method.

Define the integral error as

$$\epsilon_P(f) = \left|\hat{I}_P(f) - I(f)\right|. \tag{8}$$

Then, based on the law of large numbers, it can be easily obtained that the convergence rate of MC approximation error $\epsilon_P(f)$ is $O(L^{-1/2})$ [30,31].

In QMC methods, the low-discrepancy sequence is used to reduce the approximation error, instead of randomly sampling pseudorandom sequence. Fig. 4 shows the difference between these two sequences. There is an undesired clustering of points in the pseudorandom sequence, while the points from Halton sequence are more evenly distributed. Intuitively, by avoiding such non-uniformity, QMC methods might achieve better approximation capability. In the following, we will show that better SSP capability can be achieved by using QMC sequences instead of MC ones when generating the input weights of ELM.

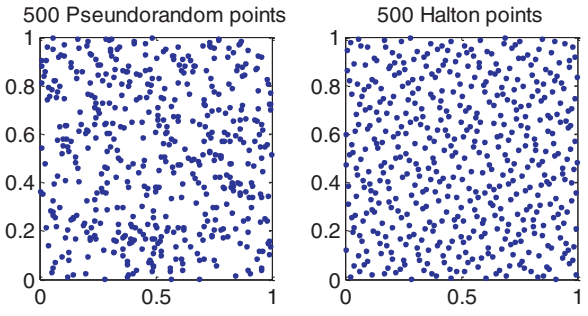

Fig. 4. Comparison of MC and QMC sequences.

3.2. SSP capability of QMC projection

Firstly, we give the classical results in QMC theory.

Lemma 1. *(Koksma–Hlawka inequality [32]) For any function f with bounded variation, and sequence $P = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L\}$, the integration error is bounded as follows,*

$$\left|\frac{1}{L}\sum_{i=1}^{L} f(\mathbf{w}_i) - \int_{[0,1]^n} f(\mathbf{x})d\mathbf{x}\right| \le V_{HK}(f)D_L^*(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L) \tag{9}$$

where $V_{HK}(f)$ is the variation of f in the sense of Hardy and Krause defined in terms of the following partial derivatives,

$$V_{HK}(f) = \sum_{k=1}^{n}\sum_{1\le i_1<i_2<\cdots\le i_k\le n} V^{(k)}(f; i_1, \ldots, i_k) \tag{10}$$

with $V^{(k)}(f; i_1, \ldots, i_k) = \int_{[0,1]^k} \left|\frac{\partial^k f}{\partial u_{i_1}\cdots\partial u_{i_k}}\right|_{u_j=1, j\ne i_1,\ldots,i_k} |du_{i_1}\ldots du_{i_k}$,

and D_L^* is the star-discrepancy defined by

$$D_L^*(P) = \sup_{\mathbf{x}\in[0,1]^n}\left|\frac{|\{i : \mathbf{w}_i \in J_\mathbf{x}\}|}{L} - \lambda_n(J_\mathbf{x})\right|$$

with $J_\mathbf{x} = [0, x_1) \times [0, x_2) \times \cdots \times [0, x_n)$, and $\lambda_n(J_\mathbf{x}) = \prod_{j=1}^{n} x_j$.

An infinite sequence $\mathbf{w}_1, \mathbf{w}_2, \ldots$ is defined as a low-discrepancy sequence if its subsequence has a low discrepancy, that is, $D_L^*(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L) = O((\log L)^n/L)$. There have been different construction method for low-discrepancy sequences, such as Halton sequences, Sobol' sequences, Faure sequences, and more [29].

Based on Lemma 1, we have the following results about the SSP capability of QMC projection.

Theorem 2. *(Convergence rate of QMC projection) Suppose $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_L\} \subset [0, 1]^{n\times 1}$ is a low-discrepancy sequence and let $\mathbf{w}_i = 2\mathbf{s}_i - 1$, then for any $\mathbf{u} \in \mathbb{R}^n$, it holds that*

$$(1-\varepsilon)\frac{L\|\mathbf{u}\|^2}{3} \le \|\mathbf{W}\mathbf{u}\|^2 \le (1+\varepsilon)\frac{L\|\mathbf{u}\|^2}{3}$$

where $\mathbf{W} = [\mathbf{w}_1^T, \mathbf{w}_2^T, \ldots, \mathbf{w}_L^T]^T$, and

$$\varepsilon = O\big((\log L)^n/L\big) \tag{11}$$

Proof. Using the fact that

$$\mathbb{E}[(\mathbf{w}\cdot\mathbf{u})^2] = \|\mathbf{u}\|^2/3,$$

if $\mathbf{w} \sim U[-1, 1]^n$, we get that

$$\|\mathbf{u}\|^2/3 = \int_{[-1,1]^n} 2^{-n}(\mathbf{w}\cdot\mathbf{u})^2 d\mathbf{w} = \int_{[0,1]^n} ((2\mathbf{s}-1)\cdot\mathbf{u})^2 d\mathbf{s}. \tag{12}$$

Suppose

$$f(\mathbf{s}) = ((2\mathbf{s}-1)\cdot\mathbf{u})^2 = \left(\sum_{i=1}^{n}(2s_i-1)\cdot u_i\right)^2. \tag{13}$$

Consider the approximation problem of the following integral

$$I(f) = \int_{[0,1]^n} f(\mathbf{s})d\mathbf{s}.$$

By Koksma–Hlawka inequality, for the given low-discrepancy sequence $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_L$, it holds that

$$\left|\frac{1}{L}\sum_{i=1}^{L} f(\mathbf{s}_i) - \int_{[0,1]^n} f(\mathbf{s})d\mathbf{s}\right| \le V_{HK}(f)D_L^*(\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_L), \tag{14}$$

where $D_L^*(\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_L) = O((\log L)^n/L)$.

Please cite this article as: W. Wang, X. Liu, The selection of input weights of extreme learning machine: A sample structure preserving point of view, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.06.079

Based on the define of f, we can calculate its variation in the sense of Hardy and Krause as follows

$$V_{HK}(f) = \sum_{1\le i_1\le n} V^{(1)}(f;i_1) + \sum_{1\le i_1,i_2\le n} V^{(2)}(f;i_1,i_2) = 8\sum_{1\le i,j\le n;i\ne j} u_i u_j \quad (15)$$

Combining Eqs. (12) and (13) and the fact that $\mathbf{w}_i = 2\mathbf{s}_i - 1$, the left hand side of Eq. (14) can be rewritten as

$$\left|\frac{1}{L}\sum_{i=1}^{L} f(\mathbf{s}_i) - \int_{[0,1]^n} f(\mathbf{s})d\mathbf{s}\right| = \left|\frac{1}{L}\sum_{i=1}^{L}((2\mathbf{s}_i-1)\cdot\mathbf{u})^2 - \frac{\|\mathbf{u}\|^2}{3}\right| = \left|\frac{1}{L}\sum_{i=1}^{L}(\mathbf{w}_i\cdot\mathbf{u})^2 - \frac{\|\mathbf{u}\|^2}{3}\right| = \left|\frac{1}{L}\|\mathbf{W}\cdot\mathbf{u}\|^2 - \frac{\|\mathbf{u}\|^2}{3}\right|. \quad (16)$$

Finally, inserting (16) into Eq. (14) and dividing it by $\|\mathbf{u}\|^2/3$, we can bound the distortion error ε as follows

$$\left|\frac{\|\mathbf{W}\cdot\mathbf{u}\|^2/L - \|\mathbf{u}\|^2/3}{\|\mathbf{u}\|^2/3}\right| \le \frac{V_{HK}(f)D_L^*(\mathbf{s}_1,\mathbf{s}_2,\ldots,\mathbf{s}_L)}{\|\mathbf{u}\|^2/3} = 24\frac{\sum_{1\le i,j\le n;i\ne j} u_i u_j}{\|\mathbf{u}\|^2} D_L^*(\mathbf{s}_1,\mathbf{s}_2,\ldots,\mathbf{s}_L) \le 24D_L^*(\mathbf{s}_1,\mathbf{s}_2,\ldots,\mathbf{s}_L) = O\left((\log L)^n/L\right)$$

This completes the proof. □

Theorem 2 shows that QMC projection can achieve a convergence rate of $O((\log L)^n/L)$ in terms of distortion errors, which is asymptotically superior to $O(1/\sqrt{L})$ for relatively small values of n. In fact, L should be exponential in n to maintain the improvement. Therefore, QMC method can be applied to relatively low-dimensional projection problem. Further analysis of the difference of the projection performance of MC and QMC methods will be conducted experimentally below.

Next we will give the method of construction of the random orthogonal projection matrix.

3.3. Random orthogonal projection and its SSP capability

Another natural way of preserving the structure similarity among samples is using orthogonal random vectors instead of direct random ones. Obviously, if the target dimension L equals to the original dimension n, any orthogonal $n \times n$ matrix can transform any sample set isometrically. In general, if $L \ne n$, we will show that good SSP performance also can be obtained by using well-designed orthogonal transform. On the one hand, if $L < n$, the information loss would be inevitable even if the orthogonal transformation is used. Nevertheless, we can try to minimize the loss of the structure similarity among samples. Theorem 3 shows that we can achieve this by using a random matrix obtained based on the principal component analysis (PCA) method.

Theorem 3. *(Construction of RO projection ($L < n$)) Given input data $\{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^n$, there exists a random matrix $\mathbf{W}_{L\times n}$, the rows of which are a set of orthonormal basis vectors, minimizing the following objective*

$$\min \sigma^2(\mathbf{W}) = \frac{1}{N^2}\sum_{i,j=1}^{N}\left(\Delta {d_{ij}}^2\right) \quad (17)$$

where $\Delta {d_{ij}}^2 = |\|\mathbf{x}_i - \mathbf{x}_j\|^2 - \|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j\|^2|$.

Proof. Without loss of generality, assume $\sum_{i=1}^N \mathbf{x}_i = \mathbf{0}$.

Noting that $\mathbf{W}$ is consisted of orthonormal vectors and $L < n$, $\|\mathbf{W}\mathbf{u}\|^2 \le \|\mathbf{u}\|^2$ holds for any $\mathbf{u} \in \mathbb{R}^n$. Hence, the objective function becomes

$$\sigma^2(\mathbf{W}) = \frac{1}{N^2}\sum_{i,j=1}^{N}\left(\|\mathbf{x}_i - \mathbf{x}_j\|^2 - \|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j\|^2\right) = \frac{1}{N^2}\left(\sum_{i,j=1}^{N}\|\mathbf{x}_i - \mathbf{x}_j\|^2 - \sum_{i,j=1}^{N}\|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j\|^2\right) = \frac{2}{N}\left(\sum_{i=1}^{N}\|\mathbf{x}_i\|^2 - \sum_{i=1}^{N}\|\mathbf{W}\mathbf{x}_i\|^2\right),$$

which actually is the loss of the empiric variance after transforming input matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ into $\mathbf{WX}$. Obviously, the truncated transformation, produced by using only the first L loading vectors corresponding to the first L principal components minimizes $\sigma^2(\mathbf{W})$. That is, if the covariance matrix $\mathbf{X}^T\mathbf{X}/N$ has a set of orthonormal eigenvectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ and a set of associated eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ (in descending order), then $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_L]^T$ is obviously one of the optimal solution.

Finally, the random orthonormal transformation can be gotten by

$$\mathbf{W} = \mathbf{R}_{L\times L}\mathbf{V},$$

where $\mathbf{R}_{L\times L}$ is a random orthogonal matrix. In practice, $\mathbf{R}_{L\times L}$ can be obtained by orthogonalizing a random $L \times L$ matrix generated by MC sampling. Because of the orthogonality of $\mathbf{R}_{L\times L}$, $\mathbf{W}$ also minimizes the empiric variance loss. □

On the other hand, if $L \ge n$, Theorem 4 shows that a projection matrix extracted from any random orthogonal matrix can accomplish the transformation without any loss of the geometrical similarity among samples.

Theorem 4. *(Construction of RO projection ($L \ge n$)) If $\mathbf{W}_{L\times n}$ ($L \ge n$) is an arbitrary matrix consisted of n orthonormal random basis vectors, then for any $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{W}\mathbf{x}\|^2 = \|\mathbf{x}\|^2$.*

Proof. $\|\mathbf{W}\mathbf{x}\|^2 = \mathbf{x}^T\mathbf{W}^T\mathbf{W}\mathbf{x} = \mathbf{x}^T\mathbf{I}_n\mathbf{x} = \|\mathbf{x}\|^2$. □

The analysis above tells that we can always construct a random orthogonal projection matrix providing the best SSP capability regardless of the size relationship between L and n. Algorithm 1 summarizes the construction method of the RO projection matrix.

Additionally, Fig. 5 illustrates the different SSP performance of MC, QMC and RO projection methods in terms of distortion errors on Computer Activity data set. It can be seen that on such 12-dimensional data set, QMC method obtains better SSP capability than MC method, while RO method performs best. This verifies the theoretical analysis above. We will give further experimental comparison of their SSP capability in the next section, where we will also concentrate on the experimental analysis of the learning

Algorithm 1. Construction of the random orthogonal projection matrix.

Input:
Input matrix $\mathbf{X}_{n\times N}$ consisted of N n-dimensional input vectors;
Objective dimension L(define $\tilde{n} = \min\{L, n\}$);
Output:
The random orthogonal projection matrix $\mathbf{W}_{L\times n}$.
Step 1: Generate a random matrix $\mathbf{A}_{L\times\tilde{n}}$ by MC method.
Step 2: Orthogonalize $\mathbf{A}$ by column based on Gram–Schmidt orthogonalization method, and then obtain $\mathbf{A}^{orth}$.
Step 3:
(1) **If** $L < n$
Compute $\mathbf{W}^{pca}$, whose rows are consisted of the first L loading vectors corresponding to the first L principal components of input matrix $\mathbf{X}$, and let $\mathbf{W}_{L\times n} = \mathbf{A}^{orth}\mathbf{W}^{pca}$
(2) **Else**
Let $\mathbf{W}_{L\times n} = \mathbf{A}^{orth}$.
Return: random orthogonal projection matrix $\mathbf{W}_{L\times n}$.

Please cite this article as: W. Wang, X. Liu, The selection of input weights of extreme learning machine: A sample structure preserving point of view, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.06.079

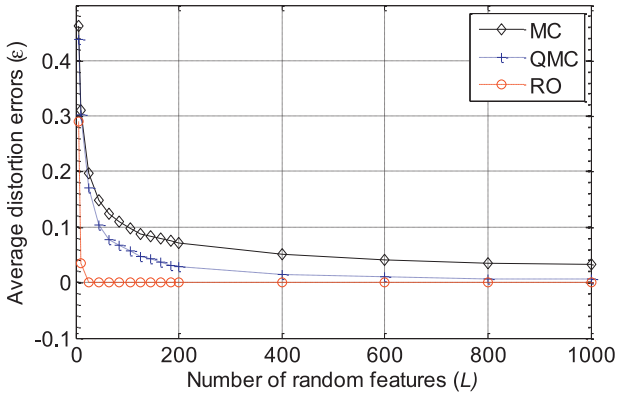

Fig. 5. The mean distortion errors of MC, QMC and RO on Computer Activity data set over 50 runs.

Table 1
Eleven benchmarking data sets.

Data sets	#Samples		#Attributes	Type
	Training	Testing		
Auto MPG	199	199	8	Regression
Machine CPU	105	104	6	Regression
Concrete Slump	82	81	10	Regression
Housing	253	253	13	Regression
Servo	84	83	4	Regression
Gisette	3500	3500	5000	Classification
Mushrooms	4062	4062	500	Classification
Sonar	207	1	60	Classification
Computer Activity	4096	4096	12	Regression
Colon Cancer	61	1	2000	Classification
Leukemia	71	1	7129	Classification

performance of ELM algorithms with QMC and RO projection methods (denoted as ELM-QMC and ELM-RO, respectively).

4. Experimental verification of ELM-QMC and ELM-RO

The experimental setup is stated in Section 4.1. The experimental results of regression and classification data sets are presented and discussed in Sections 4.2 and 4.3, respectively.

4.1. Experimental setup

Experiments of this paper are conducted on the carefully selected 6 regression and 5 classification real-world benchmarking data sets, as shown in Table 1. The data sets cover a wide range of application fields such as handwriting recognition, gene modeling, medical informatics, and so on. The input dimensions of these data sets change in a wide range from 4 to 7129. This is suitable for examining the performance of different projection methods under both $L \geq n$ and $L < n$ cases. The first eight data sets are taken from the UCI machine learning repository [33]. Because the labels of testing set of Gisette data set are not available, only the training and validation data sets are included. And the last two classification problems (see http://ntucsu.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets): Colon Cancer and Leukemia, share two interesting features: small sample size and high dimensionality of data. These two data are quite noisy because of the involving of gene expression profiles.

All inputs are scaled into [−1, 1], and outputs into [0, 1]. The data sets are randomly divided into two parts, training and testing, according to the specifications given in Table 1, which is repeated one hundred times independently during the simulations. It should be noted that the leave-one-out cross-validation (LOO-CV) strategy has been used for Colon Cancer, Leukemia and Sonar sets due to their small sample size and large dimension size. Therefore, there is no randomness in selecting the training and testing samples of these three cases. LOO-CV can yield the most reliable estimations in the presence of limited-size sample. And then the error rates (ERs) are obtained by calculating the percentage of misclassified patterns over the whole dataset.

In the implementation of three algorithms, the input weights of ELM-MC and the random matrix **A** in Algorithm 1 of ELM-RO are all generated according to the uniform distribution $U[-1, 1]$, while the QMC projection of ELM-QMC is generated based on Halton sequences. To evaluate the learning performance of ELM-MC, ELM-QMC and ELM-RO, their average training and testing root mean square errors (RMSEs) or error rates (ERs), are shown in Tables 2 and 3, in which the number of hidden nodes (L) varies at equal spacing in the ranges guaranteeing all three algorithms can obtain the optimal performance. Best testing performance for each number of hidden nodes is highlighted in bold, while the best performance for each method is underlined.

4.2. Experiments on regression data sets (low-dimensional cases)

All the regression data sets, as shown in Table 1, are relatively low-dimensional (not greater than 13). Theorem 2 tells that in such circumstances, higher convergence rate in terms of the distortion error could be obtained by replacing the MC sequences with QMC ones in ELM, while Theorems 3 and 4 indicate that the loss of the structure information among samples can be minimized or eliminated by constructing a random orthogonal projection. Fig. 6 depicts how the distortion errors of the three projection methods change with different number of random features on Machine CPU, Auto MPG and Concrete Slump data sets. Again it verifies the theoretical analysis above: QMC can obtain better SSP capability than MC, while RO always performs best.

On the other hand, experimental results in Table 2 show that ELM-QMC achieves lower testing RMSEs than ELM-MC on all six datasets except Concrete Slump case, and the corresponding relative improvements are 1.54%, 0.13%, 0.32%, −0.20%, 1.24% and 0.24%, respectively. Particularly, ELM-RO performs best among all three algorithms consistently on all six datasets. Compared with ELM-MC, the relative improvement of ELM-RO are 7.95%, 1.14%, 0.96%, 5.19%, 7.55% and 4.67%, respectively, which are obviously much more significant than those obtained by ELM-MC. Furthermore, given any number of the hidden nodes in Table 2, the optimal testing RMSEs can also be achieved by ELM-RO, except for AutoMPG ($L = 31$) and Machine CPU ($L = 7$ and 15), where ELM-QMC performs best. It is interesting to point out that, the performance differences among ELM-MC, ELM-QMC and ELM-RO found from the SSP point of view are completely consistent with those found from the testing RMSE point of view.

All the analysis above shows that for such low-dimensional learning tasks, QMC and RO can lead to the better SSP capability and hence the better testing performance under the ELM learning framework.

4.3. Experiments on classification data sets (higher dimensional cases)

Classification tasks here can be easily realized under the multi-dimensional regression learning framework of ELM [5,39]. Table 3 gives the experimental results on five classification problems, which are intentionally selected with higher input dimensions to evaluate the performance of ELM-QMC and ELM-RO comprehensively. Generally speaking, the number of hidden nodes would be much smaller than the input dimension under such

Please cite this article as: W. Wang, X. Liu, The selection of input weights of extreme learning machine: A sample structure preserving point of view, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.06.079

Table 2
Training and testing RMSEs (and their standard deviation in brackets) of ELM-MC, ELM-QMC and ELM-RO methods on regression problems.

Datasets	L	RMSEs of ELM-MC		RMSEs of ELM-QMC		RMSEs of ELM-RO	
		Training	Testing	Training	Testing	Training	Testing
Computer Activity	70	.0364(.0013)	.0407(.0037)	.0359(.0012)	.0402(.0041)	.0344(.0007)	**.0377(**.0036**)**
	80	.0349(.0010)	.0399(.0038)	.0346(.0008)	.0393(.0050)	.0335(.0007)	**.0370**(.0033)
	90	.0336(.0007)	.0390(.0081)	.0335(.0007)	.0391(.0058)	.0326(.0006)	**.0369**(.0032)
	100	.0331(.0007)	.0395(.0092)	.0327(.0006)	.0384(.0066)	.0319(.0006)	**.0365**(.0050)
	110	.0324(.0006)	.0392(.0059)	.0321(.0005)	.0387(.0055)	.0314(.0006)	**.0371**(.0049)
AutoMPG	15	.0755(.0048)	.0831(.0060)	.0738(.0049)	.0810(.0056)	.0726(.0051)	**.0803**(.0054)
	19	.0717(.0048)	.0806(.0050)	.0712(.0045)	.0802(.0053)	.0692(.0037)	**.0780**(.0050)
	23	.0670(.0045)	.0798(.0056)	.0673(.0044)	.0802(.0052)	.0662(.0039)	**.0792**(.0057)
	27	.0656(.0041)	.0789(.0053)	.0656(.0045)	.0788(.0052)	.0649(.0039)	**.0783**(.0052)
	31	.0626(.0045)	.0808(.0060)	.0625(.0043)	**.0801**(.0061)	.0624(.0041)	**.0801**(.0054)
Machine -CPU	7	.0466(.0119)	.0675(.0202)	.0448(.0114)	**.0646**(.0190)	.0453(.0098)	.0649(.0162)
	9	.0383(.0080)	.0623(.0206)	.0375(.0076)	.0621(.0195)	.0379(.0075)	**.0617**(.0210)
	11	.0335(.0066)	.0643(.0245)	.0333(.0063)	.0635(.0250)	.0330(.0062)	**.0624**(.0234)
	13	.0303(.0056)	.0673(.0291)	.0297(.0052)	.0636(.0273)	.0298(.0050)	**.0619**(.0227)
	15	.0279(.0043)	.0694(.0319)	.0277(.0044)	**.0680**(.0288)	.0278(.0046)	.0691(.0312)
Concrete Slump	80	.0783(.0029)	.1006(.0062)	.0782(.0031)	.1004(.0062)	.0743(.0037)	**.0974**(.0069)
	90	.0744(.0031)	.1002(.0065)	.0750(.0029)	.1007(.0058)	.0700(.0035)	**.0962**(.0086)
	100	.0707(.0030)	.1009(.0077)	.0707(.0029)	.1004(.0078)	.0658(.0028)	**.0957**(.0098)
	110	.0677(.0029)	.1013(.0098)	.0681(.0030)	.1015(.0091)	.0625(.0026)	**.0950**(.0086)
	120	.0641(.0028)	.1026(.0119)	.0648(.0028)	.1029(.0114)	.0590(.0024)	**.0962**(.0107)
Housing	58	.0716(.0035)	.0913(.0188)	.0713(.0035)	.0908(.0185)	.0666(.0032)	**.0844**(.0172)
	60	.0710(.0035)	.0900(.0180)	.0707(.0035)	.0888(.0163)	.0661(.0027)	**.0831**(.0159)
	62	.0698(.0037)	.0894(.0168)	.0696(.0034)	.0890(.0174)	.0654(.0027)	**.0835**(.0173)
	64	.0685(.0036)	.0887(.0182)	.0690(.0034)	.0876(.0173)	.0642(.0029)	**.0820**(.0166)
	66	.0677(.0033)	.0914(.0202)	.0676(.0034)	.0900(.0200)	.0634(.0027)	**.0847**(.0175)
Servo	25	.0787(.0134)	.1255(.0202)	.0781(.0131)	.1253(.0186)	.0712(.0120)	**.1196**(.0255)
	27	.0749(.0131)	.1243(.0199)	.0751(.0128)	.1243(.0184)	.0680(.0112)	**.1186**(.0302)
	29	.0714(.0127)	.1255(.0329)	.0708(.0125)	.1240(.0219)	.0653(.0119)	**.1185**(.0264)
	31	.0668(.0128)	.1274(.0222)	.0674(.0131)	.1271(.0202)	.0611(.0124)	**.1217**(.0308)
	33	.0645(.0124)	.1262(.0234)	.0643(.0124)	.1264(.0243)	.0597(.0111)	**.1218**(.0330)

Table 3
Training and testing error rates (and their standard deviation in brackets) of ELM-MC, ELM-QMC and ELM-RO methods on classification problems.

Datasets	L	Error rates of ELM-MC		Error rates of ELM-QMC		Error rates of ELM-RO	
		Training	Testing	Training	Testing	Training	Testing
Colon Cancer	8	.2526(.0550)	.3422(.0706)	.2529(.0490)	.3410(.0598)	.1610(.0456)	**.2190**(.0573)
	12	.1996(.0544)	.3181(.0720)	.2040(.0594)	.3276(.0793)	.1292(.0327)	**.2145**(.0482)
	16	.1600(.0476)	.3071(.0678)	.1677(.0542)	.3263(.0670)	.1098(.0289)	**.2165**(.0427)
	20	.1251(.0405)	.3020(.0633)	.1317(.0445)	.3142(.0696)	.0952(.0234)	**.2357**(.0413)
	24	.1052(.0437)	.3098(.0627)	.1067(.0414)	.3195(.0666)	.0771(.0270)	**.2364**(.0492)
Leukemia	8	.2658(.0540)	.3445(.0666)	.2619(.0452)	.3442(.0547)	.1451(.0506)	**.1960**(.0612)
	16	.1801(.0498)	.3128(.0694)	.1935(.0444)	.3266(.0575)	.0678(.0285)	**.1456**(.0446)
	24	.1165(.0436)	.3020(.0639)	.1223(.0390)	.3019(.0608)	.0410(.0237)	**.1511**(.0479)
	32	.0700(.0343)	.2972(.0641)	.0618(.0330)	.2837(.0582)	.0217(.0163)	**.1679**(.0510)
	40	.0365(.0265)	.3103(.0713)	.0308(.0275)	.2836(.0722)	.0106(.0113)	**.1919**(.0541)
Sonar	30	.1911(.0281)	.2773(.0336)	.1794(.0279)	.2674(.0345)	.1534(.0157)	**.2534**(.0180)
	50	.1275(.0221)	.2561(.0297)	.1198(.0232);	**.2447**(.0323)	.1239(.0128)	.2484(.0167)
	70	.0857(.0192)	.2532(.0300)	.0760(.0208)	.2405(.0324)	.0637(.0155)	**.2226**(.0279)
	90	.0507(.0159)	.2534(.0340)	.0429(.0143)	.2415(.0382)	.0337(.0122)	**.2172**(.0262)
	110	.0258(.0122)	.2589(.0369)	.0218(.0114)	.2497(.0333)	.0134(.0080)	**.2179**(.0290)
Gisette	450	.0870(.0034)	.1102(.0099)	.0810(.0042)	.1050(.0127)	.0255(.0019)	**.0313**(.0037)
	550	.0681(.0073)	.0932(.0079)	.0705(.0042)	.0922(.0104)	.0226(.0012)	**.0307**(.0039)
	650	.0590(.0040)	.0848(.0058)	.0582(.0055)	.0880(.0090)	.0176(.0014)	**.0272**(.0042)
	750	.0475(.0044)	.0760(.0076)	.0479(.0027)	.0800(.0084)	.0135(.0009)	**.0252**(.0039)
	850	.0429(.0025)	.0800(.0063)	.0406(.0042)	.0802(.0089)	.0114(.0017)	**.0213**(.0014)
Mushrooms	200	.0004(.0003)	.1752(.0698)	.0004(.0003)	.1778(.0948)	.0005(.0002)	**.0868**(.0104)
	280	.0001(.0002)	.1523(.0864)	.0002(.0003)	.1734(.1060)	.0000(.0001)	**.0534**(.0093)
	360	.0000(.0000)	.1505(.0824)	.0000(.0001)	.1268(.0678)	.0000(.0000)	**.0508**(.0118)
	440	.0000(.0000)	.1240(.0652)	.0000(.0000)	.1233(.0580)	.0000(.0000)	**.0554**(.0142)
	520	.0000(.0000)	.0761(.0438)	.0000(.0000)	.1119(.0334)	.0000(.0000)	**.0558**(.0175)
	600	.0000(.0000)	.0941(.0419)	.0000(.0000)	.1033(.0646)	.0000(.0000)	**.0565**(.0138)

situation, and hence the loss of structure information of samples will be inevitably during the projection stage of the learning process. It should be pointed out that the experiments do not address feature selection or feature extraction. Here we will utilize this situation to analyze in depth the practical effect of different projection methods on the SSP capability and learning performance for high-dimensional cases.

Firstly, Fig. 7 shows the mean distortion errors of MC, QMC and RO methods on Conlon Cancer, Leukemia and Sonar cases, the input dimension of which are 2000, 7129 and 60, respectively. It can be seen that for such higher dimensional problem, the QMC method can only obtain a degenerated SSP performance that is similar to or worse than that of MC method. On the contrary, the RO method can still achieve the minimal structure loss on all these

Please cite this article as: W. Wang, X. Liu, The selection of input weights of extreme learning machine: A sample structure preserving point of view, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.06.079

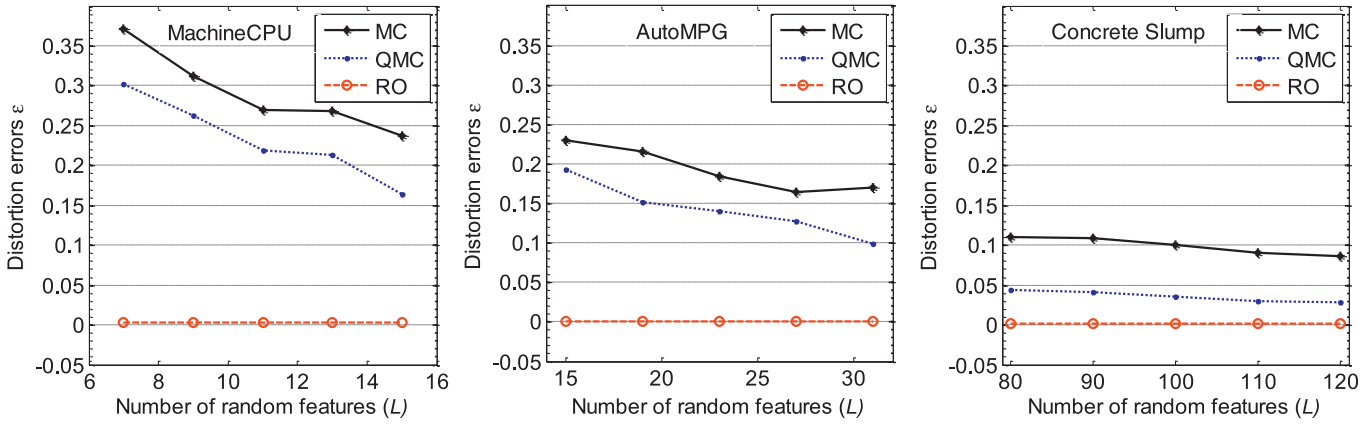

Fig. 6. The mean distortion errors of MC, QMC and RO on 3 regression data sets over 50 runs.

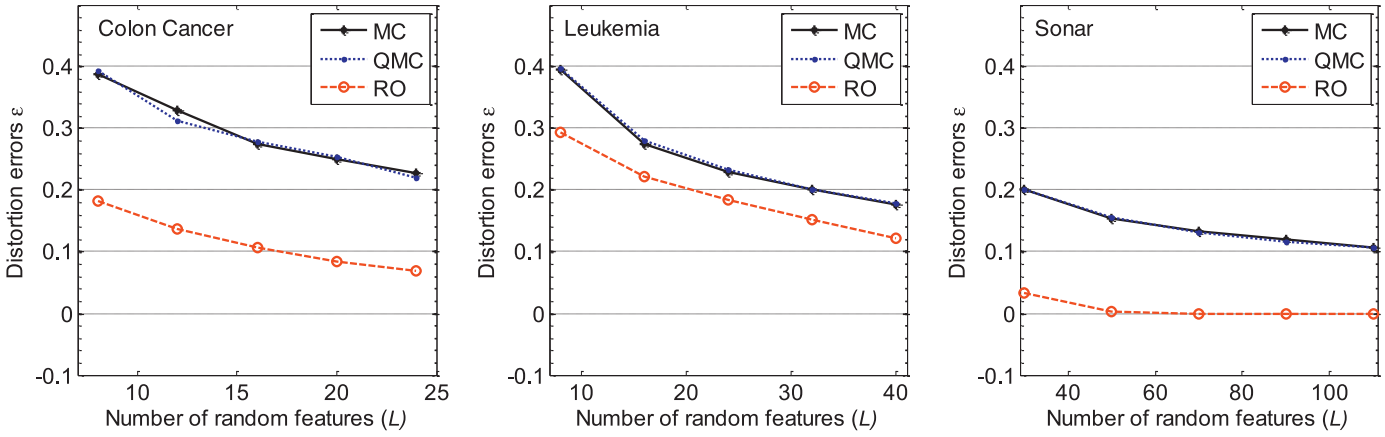

Fig. 7. The mean distortion errors of MC, QMC and RO on 3 classification data sets over 50 runs.

datasets. Obviously, this also verifies the theoretical results experimentally.

Secondly, as can be seen in Table 3, great advantages over ELM-MC can also be found in ELM-RO, and the corresponding relative improvements on testing RMSEs on the five data sets are 28.97%, 51.01%, 48.34%, 66.84% and 33.25% respectively, which are much larger than those of the low-dimensional cases. Nevertheless the advantages of ELM-QMC have been no longer in existence, since ELM-QMC outperforms ELM-MC only on Leukemia and Sonar data sets. The reason may lie in the fact that the convergence rate of distortion errors of QMC projection method varies exponentially with respect to the input dimension, which will increase very fast as the input dimension becomes large.

These experimental results show that for high-dimensional cases, it is of greater importance for ELM to improve the SSP capability when projecting the original data into a new space of much lower dimension. To this end, RO projection method should be preferred for high-dimensional applications.

5. Discussion

In this section, we provide more insights about the main works of this paper by reminding some closely related existing works, showing the connection and difference between them.

5.1. Learning with QMC method

In the works of Yang et al. [30] and Avron et al. [31], the authors have proposed to use QMC feature maps instead of MC ones for approximating the shift-invariant kernels. They show that in the case of Fourier features, QMC can reduce the error for approximating the shift-invariant kernels. But the validity of extending this result from Fourier series to other kinds of activation functions of ELM is still open. In contrast, the SSP capability of QMC proposed in our paper is obviously irrelevant to the kinds of activation functions.

On the other hand, it is a novel and more intuitive perspective to analyze the SSP capability of QMC projection. This point of view makes the classical results (e.g. Koksma–Hlawka inequality) of QMC method applicable and the theoretical analysis more direct, while in [30,31] a new discrepancy measure, called box discrepancy, had to be defined because the variation of the deduced integrand corresponding to the shift-invariant kernel is not bounded.

5.2. Learning with orthogonalization method

In previous works [34–37], the orthogonalization technique has been used in ELM. In [34,35], the orthogonalization operation are conducted on the output matrix of hidden layer of ELM not on the input weights. In [36,37], the random input weight matrix was orthogonalized when establishing the ELM auto-encoder and local receptive fields based ELM models, which had led to improved performance in their simulations. However, the orthogonalization method therein cannot guarantee its optimal SSP capability in the case of $L < n$. More importantly, the essence behind the orthogonal transformation has not been investigated in details. By contrast, this paper shows that it might be the better SSP performance that makes RO projection outperform MC projection.

In addition, PCA method has recently been used by Castano et al. [38] to select the input weights of ELM, where the input

Please cite this article as: W. Wang, X. Liu, The selection of input weights of extreme learning machine: A sample structure preserving point of view, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.06.079

weights are defined as principal components deterministically. It should be noted that the method proposed therein can only obtain the ELM networks whose sizes are not larger than the dimension of original sample space, and such deterministic approach would result in poor accuracy especially for noisy data [34]. On the contrary, the orthogonalization method proposed in our paper is a unified method that not only minimizes the loss of all distances between samples, but also maintains the randomness.

6. Conclusions and future works

In this paper, we have shown the SSP capability of different projection methods for generating input weights of ELM, and analyzed the learning performance of ELM with these projection methods.

Theoretical analysis shows that although ELM can serve as a universal approximator, the MC sampling method that is adopted for generating input weights owns poor capability of preserving the structure similarity. We show that with QMC sequences, the distortion error can converge to zero at a rate of $O((\log L)^n/L)$, which is faster than that of MC method, $O(1/\sqrt{L})$. While, the proposed RO method can provide the optimal projection matrix in the sense of the minimization of structure loss among samples. The experimental results on eleven real-world data sets show that imposing the sample structure preserving capability into ELM is an effective way to improve its learning and predicting performance.

Although this paper is mainly focused on the global SSP performance of the input weights, its locality SSP performance may be much more essential, which should be considered in the future work. Another future work could be the applications of proposed methods to more complicated learning problems such as image identification, natural language processing, and text mining.

Acknowledgments

This work was supported by Zhejiang Provincial Natural Science Foundation of China (Grant No. LY14F030020).

References

[1] G.J. Gibson, Exact classification with two-layer neural nets, J. Comput. Syst. Sci. 52 (1996) 349–356.
[2] K. Hornik, M. Stinchcombe, H. White, Multi-layer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359–366.
[3] D.E. Rumelhart, G.E. Hinton, R.J. Wiliams, Learning representations by back--propagating errors, Nature 323 (1986) 533–536.
[4] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.
[5] G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern. Part B Cybern. 42 (2) (2012) 513–529.
[6] F. Scarselli, A.C. Tsoi, Universal approximation using feedforward neural networks: a survey of some existing methods, and some new results, Neural Netw. 11 (1998) 15–37.
[7] G.B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (2006) 879–892.
[8] C.R. Rao, S.K. Mitra, Generalized Inverse of Matrices and Its Applications, Wiley, New York, 1972.
[9] A. Akusok, K.M. Bjork, Y. Miche, A. Lendasse, High performance extreme learning machines: a complete toolbox for big data applications, IEEE Access 03 (2015) 1011–1025.
[10] C.W. Deng, G.B. Huang, J. Xu, J.X. Tang, Extreme learning machines: new trends and applications, Sci. China (Inf. Sci.) 02 (2015) 5–20.
[11] X. Liu, S.B. Lin, J. Fang, Z.B. Xu, Is extreme learning machine feasible? a theoretical assessment (Part I), IEEE Trans. Neural Netw. Learn. Syst. 26 (2015) 7–20.
[12] S.B. Lin, X. Liu, J. Fang, Z.B. Xu, Is extreme learning machine feasible? a theoretical assessment (Part II), IEEE Trans. Neural Netw. Learn. Syst. 26 (2015) 21–34.
[13] R. Wang, S. Kwong, X.Z. Wang, A study on random weights between input and hidden layers in extreme learning machine, Soft Comput. 16 (2012) 1465–1475.
[14] D. Wang, P. Wang, Y. Ji, An oscillation bound of the generalization performance of extreme learning machine and corresponding analysis, Neurocomputing 151 (2015) 883–890.
[15] Y.G. Wang, F.L. Cao, Y.B. Yuan, A study on effectiveness of extreme learning machine, Neurocomputing 74 (2011) 2483–2490.
[16] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: optimally pruned extreme learning machine, IEEE Trans. Neural Netw. 21 (2009) 158–162.
[17] M. Han, X.X. Liu, An extreme learning machine algorithm based on mutual information variable selection, Control Decis. 29 (2014) 1576–1580.
[18] W.B. Johnson, J. Lindenstrauss, Extensions of Lipschitz maps into a Hilbert space, Contemp. Math. 26 (1984) 189–206.
[19] X.Y. Liu, C.H. Gao, P. Li, A comparative analysis of support vector machines and extreme learning machines, Neural Netw. 33 (2012) 58–66.
[20] Y. Zhou, B. Liu, S.X. Xia, B. Liu, Semi-supervised extreme learning machine with manifold and pairwise constraints regularization, Neurocomputing 149 (2015) 180–186.
[21] B. Liu, S.X. Xia, F.R. Meng, Y. Zhou, Manifold regularized extreme learning machine, Neural Comput. Appl. (2015) 1–15.
[22] G. Huang, S. Song, J.N. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Trans. Cybern. 44 (2014) 2405–2417.
[23] N. William, J. Anderson, T.D. Morley, Eigenvalues of the Laplacian of a graph, Linear Multilinear Algebra 18 (1985) 141–145.
[24] W. Mao, Y. Zheng, X.X. Mu, J.W. Zhao, Uncertainty evaluation and model selection of extreme learning machine based on Riemannian metric, Neural Comput. Appl. 24 (2014) 1613–1625.
[25] C. Cortes, V. Vapnik, Support vector networks, Mach. Learn. 20 (1995) 273–297.
[26] R.E. Schapire, The boosting approach to machine learning: an overview, Nonlinear Estimation and Classification, Springer, New York, 2003, pp. 149–171.
[27] L.K. Jones, A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training, Ann. Stat. 20 (1992) 608–613.
[28] R.E. Caflisch, Monte Carlo and quasi-Monte Carlo methods, Acta Numer. 7 (1997) 1–49.
[29] J. Dick, F.Y. Kuo, I.H. Sloan, High-dimensional integration: the quasi-Monte Carlo way, Acta Numer. 22 (2013) 133–288.
[30] J.Y. Yang, V. Sindhwani, H. Avron, M. Mahoney, Quasi-Monte Carlo feature maps for shift-invariant kernels, in: Proceedings of the Thirty-first International Conference on Machine Learning, 2014, pp. 485–493.
[31] H. Avron, V. Sindhwani, J.Y. Yang, M. Mahoney, Quasi-Monte Carlo feature maps for shift-invariant kernels, arXiv:1412.8293, 2014.
[32] L. Brandolini, L. Colzani, G. Gigante, G. Travaglini, On the Koksma-Hlawka inequality, J. Complexity 29 (2) (2013) 158–172.
[33] K. Bache, M. Lichman, UCI machine learning repository, 2013. URL ⟨http://archive.ics.uci.edu/ml (accessed 2015.05.21).
[34] N. Wang, M.J. Er, M. Han, Parsimonious extreme learning machine using recursive orthogonal least squares, IEEE Trans. Neural Netw. Learn. Syst. 25 (2014) 1828–1841.
[35] Y.P. Zhao, K.K. Wang, Y.B. Li, Parsimonious regularized extreme learning machine based on orthogonal transformation, Neurocomputing 156 (2015) 280–296.
[36] G.B. Huang, Z. Bai, L.L.C. Kaun, C.M. Vong, Local receptive fields based extreme learning machine, IEEE Comput. Intell. Mag. 10 (2015) 18–29.
[37] L.L.C. Kasun, H. Zhou, G.B. Huang, C.M. Vong, Representational learning with extreme learning machine for big data, IEEE Intell. Syst. 28 (2013) 31–34.
[38] A. Castano, F. Fernández-Navarro, C. Hervás-Martínez, PCA-ELM: a robust and pruned extreme learning machine approach based on principal component analysis, Neural Process. Lett. 37 (2013) 377–392.
[39] W.T. Mao, S.J. Zhao, X.X. Mu, H.C. Wang, Multi-dimensional extreme learning machine, Neurocomputing 149 (2015) 160–170.

Wen-hui Wang received the B.Sc. degree in school of mathematical sciences from Shandong Normal University, China in 2002 and the M.Sc. degree in school of mathematical sciences from Zhejiang University, China in 2005. Her research interests include neural networks, machine learning and data mining. She has published a number of papers in international journals and conferences.

Xue-Yi Liu received the M.Sc. degree in school of mathematical sciences from Zhejiang University, China in 2005 and the Ph.D. degree in school of aeronautics and astronautics from Zhejiang University, China in 2013. Since September 2013, he has been an Associate Professor at China Jiliang University, China. His research interests include neural networks, machine learning and data mining.

Please cite this article as: W. Wang, X. Liu, The selection of input weights of extreme learning machine: A sample structure preserving point of view, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2016.06.079