Brief Papers

Low-Discrepancy Points for Deterministic Assignment of Hidden Weights in Extreme Learning Machines

Cristiano Cervellera and Danilo Macciò

***Abstract*— The traditional extreme learning machine (ELM) approach is based on a random assignment of the hidden weight values, while the linear coefficients of the output layer are determined analytically. This brief presents an analysis based on geometric properties of the sampling points used to assign the weight values, investigating the replacement of random generation of such values with low-discrepancy sequences (LDSs). Such sequences are a family of sampling methods commonly employed for numerical integration, yielding a more efficient covering of multidimensional sets with respect to random sequences, without the need for any computationally intensive procedure. In particular, we prove that the universal approximation property of the ELM is guaranteed when LDSs are employed, and how an efficient covering affects the convergence positively. Furthermore, since LDSs are generated deterministically, the results do not have a probabilistic nature. Simulation results confirm, in practice, the good theoretical properties given by the combination of ELM with LDSs.**

***Index Terms*— Discrepancy, extreme learning machines (ELMs), low-discrepancy sequences (LDSs), universal approximation.**

I. Introduction

Extreme learning machines (ELMs) are a framework for neural network training in which the values of the weights in the hidden layer are randomly assigned, and only the linear coefficients of the output layer are determined analytically. Depending on the kind of chosen algorithm, the output weights are typically determined either by least-squares pseudoinversion or by an iterative procedure. ELMs, nowadays, are supported by a rich literature (for an introduction on ELM algorithms see [1]–[4] and and the references therein) and are successfully employed in different applications [5]–[7].

A key reason for the popularity of this kind of networks is that they are characterized by a very small computational burden, since they avoid the training of the hidden layer. Yet, they are proved to retain important properties of classic neural networks, such as the universal approximation capability [8].

Lately, the possibility of replacing the pure random assignment of hidden units in the ELM context has attracted researchers. For instance, PCA has been proposed for classification tasks [9] and the use of pattern points has been analyzed for kernel versions of the ELM [4]. Here, we investigate the possibility of replacing the random assignment with a deterministic one, for the classic ELM algorithm in which the feature map is preassigned. The aim is twofold: 1) to obtain results that are not subject to a probabilistic confidence and 2) possibly to improve the performance by choosing smart algorithms for the placement of the points.

In this brief, the universal approximation capabilities of ELM are restated from a geometric point of view. In particular, the analysis relies on the concept of the discrepancy of a set of sampling points. This measure, commonly employed for numerical integration and number-theoretic methods (see [10], [11]), quantifies how uniformly a set of points covers a given multidimensional region. It will be proved that the universal approximation capability can be guaranteed when the discrepancy converges to zero, and how a faster convergence affects the approximation convergence positively.

This will allow to introduce and investigate the use of low-discrepancy point sets and (t, n)-sequences, deterministic sampling algorithms specifically aimed at yielding point sets with low-discrepancy and good convergence rates for the discrepancy itself [12]. In particular, since the convergence rate of these algorithms can be proved to be better than the one provided by sets obtained through random sampling, their use should, from a theoretical point of view, yield better results with respect to pure random sampling for the assignment of the hidden weight values.

From a practical point of view, no intensive procedure is required to generate the sampling points, which makes their use equivalent, computationally, as employing a pseudorandom number generator for the standard ELM implementation. LDSs are generated through very simple expressions, and they can be found already implemented in most commonly employed numerical software packages.

Low-discrepancy sampling has already been proved to yield good results in machine learning and approximate dynamic programming problems for the generation of input samples, when there is freedom to sample the input space (see [13], [14]). LDSs have also been proved to yield good results when they are employed in local approximation based on kernel smoothing [15].

The reasons to consider LDSs as good alternatives to pure random sampling in ELM algorithms, as well as in other contexts, are summarized as follows.

1) They are very simple to generate, and already implemented in most commonly employed numerical software packages.
2) They are endowed with better convergence bounds with respect to independent identically distributed (i.i.d.) sequences.
3) The convergence bounds are not probabilistic.

The analysis here is carried out for learning machines having the feedforward one-hidden-layer structure with analytic activation function (which includes popular functions such as the logistic function and the hyperbolic tangent), but the use of LDSs can be easily extended to other kinds of structures, such as, e.g., radial basis functions.

Then, to test the use of LDSs in practice, simulation results involving different kinds of data sets are presented, considering both the standard and the iterative version of the ELM training algorithm. The results confirm that low-discrepancy sets can be employed in practice as efficient alternatives to random sampling for the assignment of the hidden weights.

II. Learning Framework

The addressed problem concerns finding a functional dependence $f: X \to \mathbb{R}$, where X is a compact subset in the d-dimensional space $\mathbb{R}^d$, starting from a finite set of observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L\}$. We assume that the target function f belongs to the set of square integrable functions $L^2(X)$ that satisfy $\int_X |f(\boldsymbol{x})|^2 \, d\boldsymbol{x} < \infty$. We consider the usual norm and inner product defined in $L^2(X)$,

Manuscript received September 19, 2014; accepted April 19, 2015. Date of publication May 7, 2015; date of current version March 15, 2016.

The authors are with the Institute of Intelligent Systems for Automation, National Research Council, Genoa 16149, Italy, (e-mail: cristiano.cervellera@cnr.it; danilo.maccio@cnr.it).

Digital Object Identifier 10.1109/TNNLS.2015.2424999

that is, $\|f\|^2 = \int_X |f(\boldsymbol{x})|^2 d\boldsymbol{x}$ and $\langle f, g\rangle = \int_X f(\boldsymbol{x})g(\boldsymbol{x})\, d\boldsymbol{x}$ for every $f, g \in L^2(X)$.

The approximation of the objective function f is sought inside the class of one-hidden layer feedforward neural networks with N neural units, activation function $\sigma : \mathbb{R} \to \mathbb{R}$, and hidden weights bounded by B. We define such a class of functions as $\Gamma(\sigma, B) = \{g_N(\boldsymbol{x}) = \sum_{j=1}^{N} c_j \sigma(\boldsymbol{\alpha}_j^{\mathsf{T}}\boldsymbol{x} + \beta_j) : \max_{1\leqslant j\leqslant N}\{|\boldsymbol{\alpha}_j|, |\beta_j|\} \leqslant B\}$, where $|\cdot|$, applied to a vector $\boldsymbol{x} = (x_1, \ldots, x_d)^{\mathsf{T}}$, denotes the maximum norm: $|\boldsymbol{x}| = \max_{1\leqslant j\leqslant d} |x_j|$.

Given an activation function $\sigma(\boldsymbol{\alpha}_j^{\mathsf{T}}\boldsymbol{x} + \beta_j)$, we denote as $\boldsymbol{w}_j \in W$, the column vector composed by the hidden weights, that is, $\boldsymbol{w}_j = \binom{\boldsymbol{\alpha}_j}{\beta_j}$. If $\Psi = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots\}$ is an infinite sequence of weights, we define $\sigma_j(\boldsymbol{x}) = \sigma(\boldsymbol{\alpha}_j^{\mathsf{T}}\boldsymbol{x} + \beta_j)$, for $j = 1, 2, \ldots$ Then, we have $g_N(\boldsymbol{x}) = \sum_{j=1}^{N} c_j \sigma_j(\boldsymbol{x})$.

It is worth noting that the weights $\boldsymbol{w}_j$, $j = 1, 2, \ldots$, of the activation functions of every $g_N \in \Gamma(\sigma, B)$, are contained in a $d+1$-dimensional hypercube $W \subset \mathbb{R}^{d+1}$ of radius B. This is required for the analysis presented in the following, and consistent with practical implementation of ELMs, where the values are typically bounded between -1 and 1 [8].

We provide two definitions that will be useful in the rest of the analysis.

Definition 1: We say that a function $\sigma : \mathbb{R} \to \mathbb{R}$ is analytic at $\xi \in \mathbb{R}$ with radius of convergence $r > 0$ if there is an infinite sequence of real numbers $\{\gamma_j\}$, $j = 0, 1, \ldots$, such that, for $|z - \xi| < r$, the series $\sum_{j=0}^{\infty} \gamma_j (z - \xi)^j$ converges and $\sigma(z) = \sum_{j=0}^{\infty} \gamma_j (z - \xi)^j$. Furthermore, if $\gamma_j \neq 0$ for every $j \geqslant 1$, then we say that σ is superanalytic at a ξ with radius of convergence $r > 0$.

Definition 2: We say that $\Gamma(\sigma, B)$ is uniformly dense on $C(X)$ if, for every $f \in C(X)$ and $\varepsilon > 0$, there exists $g \in \Gamma(\sigma, B)$, such that $\sup\{|f(\boldsymbol{x}) - g(\boldsymbol{x})| : \boldsymbol{x} \in X\} < \varepsilon$. We say that $\Gamma(\sigma, B)$ is uniformly dense on *compacta* in $C(\mathbb{R}^d)$ if it is uniformly dense on $C(X)$ for every compact set $X \subset \mathbb{R}^d$.

III. Overview of Discrepancy and Low-Discrepancy Sequences

The discrepancy of a set of points is a measure of its uniformity over a multidimensional compact set. Define $n = d + 1$ as the size of the space W of the hidden weights that needs to be sampled for the generation of the weight values. The definition and the analysis of the discrepancy are given on the n-dimensional unitary hypercube $[0, 1]^n$. This is not a limitation, since the space W of the hidden units is a hypercube (see Section II) and one can extend the results presented in this section by simple scaling [10].

Given a set of N points $\Psi = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N\}$ in $[0, 1]^n$, let ζ be the family of all subintervals B of the form $\prod_{i=1}^{n}[a_i, b_i]$, where $a_i, b_i \in [0, 1]$, and let $A(B, \Psi)$ be the counting function for the number of points of Ψ that belong to B. Then, the discrepancy of Ψ is defined as follows:

$$D_N(\Psi) = \sup_{B\in\zeta} \left| \frac{A(B, \Psi)}{N} - \lambda(B) \right| \tag{1}$$

where $\lambda(B)$ is the Lebesgue measure of B.

In the case, where $\Psi = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots\}$ is an infinite sequence of points, the discrepancy D_N of Ψ is defined considering the first N points of the sequence, that is, $D_N(\Psi) = D_N(\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N\})$.

According to the definition, the discrepancy measures the uniformity of the points of Ψ over W in the following sense: if we subdivide W into a number of basic subsets, each one should contain a number of points that is as proportional as possible to the volume of the subset itself.

In the past years, much research effort has been put to develop algorithms to generate sequences of points that implement this concept in a deterministic and efficient way. This has led to the development of a family of sequences called low-discrepancy sequences (LDSs). An LDS aims at keeping the discrepancy of the resulting points in $[0, 1]^n$ as small as possible, and provides a favorable asymptotical rate of convergence of the discrepancy itself. Examples of such sequences are the Sobol', the Niederreiter, the Halton, and so on [12]. The concept of LDS has been generalized with the introduction of (t, n)-sequences. To illustrate their basics, the definition of a (t, q, n)-net is first provided.

Definition 3: An elementary interval in base b (where $b \geqslant 2$ is an integer) is a subinterval E of $[0, 1]^n$ having the form $E = \prod_{i=1}^{n}[a_i b^{-p_i}, (a_i + 1)b^{-p_i})$, where $a_i, p_i \in \mathbb{Z}$, $p_i > 0$, $0 \leqslant a_i \leqslant b^{p_i}$ for $1 \leqslant i \leqslant n$.

Let t, q be two integers satisfying $0 \leqslant t \leqslant q$. A (t, q, n)-net in base b is a set F of b^q points in $W \subset \mathbb{R}^n$ such that $A(E, F) = b^t$ for every elementary interval E in base b with $\lambda(E) = b^{t-q}$.

A (t, q, n)-net is clearly endowed with the property of good uniform spreading in W in the sense mentioned above. If the sample Ψ is a (t, q, n)-net in base b, every elementary interval in which we divide $[0, 1]^n$ must contain b^t points of Ψ. However, since the cardinality of a (t, q, n)-net has to be equal to b^q, (t, n)-sequences are derived from nets to provide a higher degree of freedom in choosing the number N of sampling points.

Definition 4: Let $t \geqslant 0$ be an integer. A sequence $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots,\}$ of points in $[0, 1]^n$ is a (t, n)-sequence in base b if, for all the integers $k \geqslant 0$ and $q \geqslant t$, the point set consisting of $\{\boldsymbol{w}_{kb^q}, \ldots, \boldsymbol{w}_{(k+1)b^q}\}$ is a (t, q, n)-net in base b.

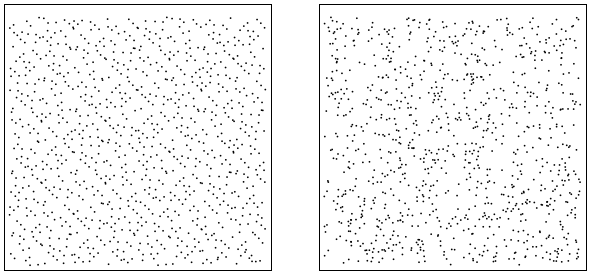
In practice, the construction of a (t, q, n)-net in base b having $N = b^q$ points can be obtained by first choosing n $(q \times q)$ matrices $P_1, \ldots, P_n$ with elements in $Z_b = \{0, \ldots, b - 1\}$. Then, to generate the jth point $\boldsymbol{w}_j$ of the net, write j in its b-adic expansion, i.e., $j = \sum_{i=0}^{q-1} v_i b^i$ with digits $v_i \in Z_b$, and consider the q-dimensional vector $\boldsymbol{v} = (v_0, \ldots, v_{q-1})^{\mathsf{T}}$.

Next, to generate the pth component of $\boldsymbol{w}_j$, multiply the pth matrix P_p by $\boldsymbol{v}$, which leads to the q-dimensional vector $(y_{j,p,1}, \ldots, y_{j,p,q})^{\mathsf{T}} = P_p \cdot \boldsymbol{v}$, where each element is again in Z_b. Finally, the component p of the jth point can be obtained as $w_{j,p} = y_{j,p,1}b^{-1} + y_{j,p,2}b^{-2} + \cdots + y_{j,p,q}b^{-q}$.

Notice that the more the columns of the various matrices P_p are mutually independent, the more the uniformity of the points of the net is ensured. The construction of a (t, n)-sequence is obtained in the same way, but the number of points (and, hence, the dimension of the matrices $P_1, \ldots, P_n$) is not specified *a priori*. For a detailed tractation, the reader is referred to [11], where the specific construction of popular LDSs such as those mentioned above is also presented. As seen, the algorithm for the generation of the points does not imply any computationally intensive procedure and, nowadays, it can be found already implemented, together with suitable generator matrices, for the most commonly employed numerical software platforms.

Concerning discrepancy, the fundamental property of (t, n)-sequences is that they are proved to yield a convergence rate of order $O(N^{-1}(\log N)^{n-1})$ [12]. This means that the convergence of the discrepancy, ignoring the logarithmic factor, is asymptotically almost linear with respect to the number of points N. For a comparison, it can be proved [16] that an i.i.d. sequence of N points drawn with uniform distribution yields a rate of convergence for the discrepancy of order $O(1/\sqrt{N})$. Furthermore, the latter rate is obviously not deterministic, being subject to a given probabilistic confidence.

Fig. 1 shows the sampling of the 2-D unit cube by means of 1000 samples obtained from a Sobol' LDS (left) and a uniform



(a) (b)

Fig. 1. (a) Sobol'. (b) Purely random sequence.

i.i.d. random sequence (right). It can be clearly seen how the low-discrepancy sampling scheme covers the space in a more uniform and regular way.

IV. Analysis of the Universal Approximation Property

In this section, we prove that using an LDS to sample the space of the hidden weights preserves the universal approximation capability of the network. The analysis relies on the classic proof in [8], where the universal approximation property is proved in $L^2(X)$ for hidden weight values randomly generated from a probability density function over $\mathbb{R}^n$ (recall that $n = d + 1$).

First, we prove the following lemma.

Lemma 1: If the activation function σ is superanalytic at some $\xi \in \mathbb{R}$ with radius $r > 0$, then $\Gamma(\sigma, B)$ is dense in $L^p(X)$, with $p \in [1, \infty)$, for any $B \geqslant \max\{|\xi|, 1\}$.

Proof: Let us take $f \in L^p(X)$ and $\varepsilon > 0$. It follows from standard theorems (see [17, Proposition 8.17]) that $C(X)$ is dense on $L^p(X)$. Then, we can take $f' \in C(X)$, such that $\|f - f'\|_p < \varepsilon/2$.

It is proved in [18, Th. 2.6] that $\Gamma(\sigma, B)$, with $B \geqslant \max\{|\xi|, 1\}$, is uniformly dense on *compacta* in $C(\mathbb{R}^d)$. Then, it is uniformly dense on $C(X)$, being X a compact set. Therefore, we can take $g \in \Gamma(\sigma, B)$, such that $\sup\{|f'(x) - g(x)|^p : x \in X\} < (\varepsilon/2)^p$. Since the Lebesgue measure of X is 1, we have that $\|f' - g\|_p < \varepsilon/2$.

The assertion follows directly by triangle inequality: $\|f - g\|_p \leqslant \|f - f'\|_p + \|f' - g\|_p < \varepsilon/2 + \varepsilon/2 = \varepsilon$. ■

Examples of activation functions satisfying the conditions of Lemma 1 are the sine, the cosine, the hyperbolic tangent, and the logistic function. Notice also that if the derivative of the function σ satisfies suitable hypotheses, then the denseness property is guaranteed with hidden weights belonging to the unit sphere (see [18, Th. 2.8]). This is the case, for instance, of popular sigmoidal activation functions, such as the hyperbolic tangent and the logistic function.

Given a sequence of weights $\Psi = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots\}$, we define, for the corresponding sequence of functions $\{\sigma_1, \sigma_2, \ldots\}$, the segment $G_{(m,M)} = \{\sigma_m, \sigma_{m+1}, \ldots, \sigma_{m+M-1}\}$, where $m = 1, 2, \ldots$, and M is a positive integer.

The following result plays the role of [8, Lemma II.6], and the proof follows its lines, from the point of view of the discrepancy of the sample $\Psi = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots\}$ of points in W used to assign the weight values.

Lemma 2: Assume the sequence Ψ is such that, for every integer $m \geqslant 0$, the discrepancy $D_N(\Psi_m)$, where $\Psi_m = \{\boldsymbol{w}_m, \boldsymbol{w}_{m+1}, \ldots\}$, converges to 0 as $N \to \infty$ and the activation function σ is bounded and nonconstant. Then, for any given positive value $\hat{\theta} < \pi/2$ and any given σ_0, there exists M sufficiently large such that there exists $\sigma_j \in G_{(m,M)}$ $(m = 1, 2, \ldots)$ satisfying $\theta_{(\sigma_j,\sigma_0)} < \hat{\theta}$, where $\theta_{(\sigma_j,\sigma_0)}$ denotes the angle formed by σ_j and σ_0.

TABLE I
Simulation Tests Setup

	d	# train points	# test points	D-TRAIN ν_1	D-TRAIN ν_2	D-TRAIN ν_3	I-TRAIN ν_1	I-TRAIN ν_2	I-TRAIN ν_3
ABA	7	2177	2000	8	12	16	5	10	20
AIR	5	800	703	10	30	50	20	30	50
CON	8	530	500	20	35	50	20	50	100
PRO	9	5000	5000	30	40	50	20	50	100
INV	9	3000	2000	50	150	250	50	100	150
F20	20	5000	5000	50	80	100	20	50	100
MUa	68	559	500	5	10	25	5	10	25
MUb	116	559	500	5	10	25	5	10	25
SLI	384	5000	5000	50	100	500	50	100	500

Proof: Given a positive value $\hat{\theta} < \pi/2$, according to [8, Lemma II.5], there exists $\delta \in (0, B]$ such that, for all $\boldsymbol{w} \in Q_0 = \{\boldsymbol{w} : \|\boldsymbol{w} - \boldsymbol{w}_0\| < \delta\}$, $\|\sigma - \sigma_0\| < \|\sigma_0\| \sin(\hat{\theta})$.

Given a generic integer M, we have that, by the definition of the discrepancy, any rectangular subinterval $Q_1 \subset W$ containing at most one point, among the first M points of the set Ψ_m, has Lebesgue measure bounded by

$$\lambda(Q_1) \leqslant D_M(\Psi_m) + \frac{1}{M}. \tag{2}$$

By assumption, we have that $D_M(\Psi_m) \to 0$ for $M \to \infty$, then the Lebesgue measure of Q_1 can be made arbitrarily small by augmenting M. In particular, there exists M such that $\lambda(Q_1) < \lambda(Q_0)$, and in this case, Q_0 contains at least one point of Ψ_m. Then, there exists M such that, for any m, there exists $\sigma_i \in G_{(m,M)}$ satisfying $\sin(\theta_{(\sigma_i,\sigma_0)}) \leqslant \|\sigma - \sigma_0\|/\|\sigma_0\| < \sin(\hat{\theta})$, where $\theta_{(\sigma_i,\sigma_0)} < \pi/2$, i.e., $\theta_{(\sigma_i,\sigma_0)} < \hat{\theta}$. ■

The following proposition shows that, if (t, n)-sequences are employed, the condition on the sequence Ψ stated by Lemma 2 is satisfied.

Proposition 1: Let $\Psi = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots\}$ a (t, n)-sequence in base b. Then, the sequence Ψ_m has discrepancy converging to 0 for every integer m.

Proof: Let us consider an arbitrary $q \geqslant t$. For a suitable $k \geqslant 0$, such that $kb^q \leqslant m \leqslant (k+1)b^q = k^*$, we can perform the following decomposition: $\Psi_m = \{\boldsymbol{w}_m, \ldots, \boldsymbol{w}_{k^*}\} \cup \{\boldsymbol{w}_{k^*+1}, \boldsymbol{w}_{k^*+2}, \ldots\} = \Psi' \cup \Psi''$.

The sequence of points Ψ'' is a (t, n)-sequence (see Definition 4), then $D_N(\Psi'') \to 0$, for $N \to \infty$.

It is possible to prove (see [11, Proposition 3.16]) that, for any given $N \leqslant k^* - m$

$$D_N(\Psi_m) \leqslant \frac{k^* - m}{N} D_{k^*-m}(\Psi') + \frac{N - k^* + 1}{N} D_N(\Psi'')$$

from which it follows that $D_N(\Psi_m) \to 0$ for $N \to \infty$. ■

With the previous two lemmas, we can state the universal approximation main result, which can be interpreted as the discrepancy-based version of [8, Th. II.1].

The residual error between the target function f and the approximation given by g_N is given by $e_N = \|f - g_N\|$.

Theorem 1: Given any bounded nonconstant superanalytic function σ, for any continuous target function f and any sequence of weights $\Psi = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots\}$ such that $\lim_{N\to\infty} D_N(\Psi) = 0$, we have that $\lim_{N\to\infty} \|e_N\| = 0$ holds if

$$c_N = \frac{\langle e_{N-1}, \sigma_N \rangle}{\|\sigma_N\|^2}, \qquad N = 1, 2, \ldots$$

Proof: The proof follows the lines of [8, Th. II.1] where Lemmas II.3 and II.6 are replaced by Lemmas 1 and 2 of this brief. ■

TABLE II
AVERAGE MEANS AND STANDARD DEVIATIONS (IN 10^{-1} UNITS)

		Batch			Iterative $K = 1$		
		N_1	N_2	N_3	N_1	N_2	N_3
ABA	**LD-SB**	$0.0634, 0.0020$	$0.0598, 0.0013$	$0.0587, 0.0007$	$0.1492, 0.0564$	$0.1386, 0.0470$	$0.1181, 0.0259$
	LD-NX	$0.0628, 0.0030$	$0.0597, 0.0012$	$0.0586, 0.0008$	$0.1392, 0.0464$	$0.1199, 0.0334$	$0.1038, 0.0199$
	UIID	$0.0636, 0.0034$	$0.0598, 0.0013$	$0.0588, 0.0008$	$0.1416, 0.0542$	$0.1279, 0.0428$	$0.1099, 0.0229$
AIR	**LD-SB**	$0.1710, 0.011$	$0.1241, 0.0080$	$0.0143, 0.004$	$0.3150, 0.0956$	$0.2754, 0.0652$	$0.2379, 0.0329$
	LD-NX	$0.1722, 0.011$	$0.1228, 0.0075$	$0.145, 0.0051$	$0.4439, 0.1485$	$0.3938, 0.1209$	$0.3224, 0.0809$
	UIID	$0.1731, 0.011$	$0.1246, 0.0064$	$0.149, 0.0051$	$0.4510, 0.0173$	$0.3952, 0.1335$	$0.3171, 0.0833$
CON	**LD-SB**	$0.1492, 0.014$	$0.1187, 0.010$	$0.1079, 0.011$	$0.4429, 0.0726$	$0.2971, 0.0381$	$0.2385, 0.0236$
	LD-NX	$0.1437, 0.014$	$0.1183, 0.012$	$0.1083, 0.009$	$0.4469, 0.0985$	$0.3371, 0.0533$	$0.2671, 0.0350$
	UIID	$0.1489, 0.015$	$0.1206, 0.013$	$0.1105, 0.011$	$0.4461, 0.0982$	$0.3218, 0.0517$	$0.2512, 0.0274$
PRO	**LD-SB**	$0.5892, 0.006$	$0.5825, 0.007$	$0.5801, 0.011$	$0.8309, 0.0472$	$0.7795, 0.0281$	$0.7371, 0.0193$
	LD-NX	$0.5902, 0.006$	$0.5831, 0.008$	$0.5799, 0.013$	$0.8449, 0.0615$	$0.7889, 0.0416$	$0.7456, 0.0265$
	UIID	$0.5906, 0.006$	$0.5848, 0.008$	$0.5815, 0.011$	$0.8449, 0.0630$	$0.7944, 0.0411$	$0.7501, 0.0278$
INV	**LD-SB**	$0.0251, 0.0033$	$0.00912, 0.0015$	$0.00433, 0.0006$	$0.0803, 0.0173$	$0.0445, 0.0058$	$0.0397, 0.0045$
	LD-NX	$0.0229, 0.0035$	$0.00840, 0.0017$	$0.00415, 0.0006$	$0.1275, 0.0357$	$0.0635, 0.0119$	$0.0460, 0.0050$
	UIID	$0.0241, 0.0032$	$0.00859, 0.0014$	$0.00425, 0.0004$	$0.1314, 0.0417$	$0.0609, 0.0118$	$0.0444, 0.0052$
F20	**LD-SB**	$0.0462, 0.0035$	$0.0383, 0.0013$	$0.0347, 0.0012$	$0.3173, 0.0622$	$0.1882, 0.0429$	$0.1058, 0.0210$
	LD-NX	$0.0495, 0.0035$	$0.0409, 0.0022$	$0.0370, 0.0020$	$0.3740, 0.0899$	$0.2572, 0.0656$	$0.1774, 0.0452$
	UIID	$0.0482, 0.0028$	$0.0401, 0.0020$	$0.0366, 0.0018$	$0.3653, 0.0778$	$0.2382, 0.0535$	$0.1393, 0.0295$
MUa	**LD-SB**	$0.4453, 0.0465$	$0.4085, 0.0168$	$0.3936, 0.0121$	$0.9509, 0.4019$	$0.8116, 0.3275$	$0.8103, 0.3498$
	LD-NX	$0.4647, 0.0934$	$0.4138, 0.0240$	$0.3962, 0.0157$	$1.054, 0.4901$	$0.9831, 0.4169$	$0.8465, 0.3281$
	UIID	$0.4891, 0.1061$	$0.4239, 0.0391$	$0.3977, 0.0155$	$1.040, 0.5330$	$0.9893, 0.4807$	$0.9299, 0.4134$
MUb	**LD-SB**	$0.4398, 0.0594$	$0.4155, 0.0252$	$0.3998, 0.0229$	$0.7510, 0.3099$	$0.6840, 0.2556$	$0.6429, 0.2277$
	LD-NX	$0.4499, 0.0680$	$0.4185, 0.0295$	$0.4031, 0.0378$	$0.7585, 0.3211$	$0.8249, 1.740$	$0.7953, 1.719$
	UIID	$0.4726, 0.0132$	$0.4189, 0.0219$	$0.4006, 0.0232$	$0.7887, 0.3718$	$0.7583, 0.3460$	$0.6890, 0.2982$
SLI	**LD-SB**	$0.3089, 0.0472$	$0.2387, 0.0332$	$0.1868, 0.0171$	$0.5724, 0.1259$	$0.4663, 0.1073$	$0.3269, 0.0693$
	LD-NX	$0.3090, 0.0521$	$0.2358, 0.0343$	$0.1871, 0.0187$	$0.6493, 0.1404$	$0.5177, 0.1097$	$0.3592, 0.0751$
	UIID	$0.3182, 0.0499$	$0.2399, 0.0321$	$0.1892, 0.0181$	$0.6598, 0.1614$	$0.5371, 0.1227$	$0.3634, 0.0765$

The proof of Lemma 2 points out that the performance of the overall algorithm is affected by the rate of convergence of the discrepancy of the sequence Ψ. Expression (2) shows that a smaller discrepancy entails a faster fulfillment of the condition $\lambda(Q_1) < \lambda(Q_0)$, which is essential for the proof of Lemma 2. Then, the use of LDSs is an improvement in this sense over random assignment, due to the faster convergence of the discrepancy (see Section III). Furthermore, it is worth to remark how this rate is deterministic, whereas it is obviously probabilistic in the random case.

V. SIMULATION TESTS

This section presents simulation results concerning nine test data sets. To cover a wide range of application contexts, seven data sets come from real regression problems (i.e., the abalone, the airfoil self-noise, the concrete compressive strength, the protein tertiary structure, the location of Computer Tomography slices, and two from the geographical origin of music, all available on the University of California, Irvine machine learning repository [19]). The remaining two are the final-stage cost-to-go function of an approximate dynamic programming problem (specifically, a 9-D inventory forecasting problem whose details can be found in [20]) and a 20-D function characterized by an oscillatory behavior, having the form $f(\boldsymbol{x}) = \sum_{i=1}^{20} \sin[c_i(x_i - r_i)^2]$, where $\boldsymbol{c}$ and $\boldsymbol{r}$ are vectors of real numbers.

All the input components have been normalized between -1 and 1, while the output values have been normalized between 0 and 1.

Two kinds of ELM algorithms have been considered: 1) the standard direct one (denoted as D-TRAIN in the following) in which all the hidden layer weights are assigned in one shot and 2) the improved iterative one described in [2] (denoted as I-TRAIN in the following). In the latter, a new neural unit is added at each iteration, and the corresponding weight values are chosen among a set of K candidate points by selecting the one that improves performance the most. The algorithm stops when either a maximum number of neural units is reached, or a desired level of accuracy is attained. Concerning the direct training algorithm, three different values of the number N of neural units have been considered. For all the test problems, the values have been chosen so that the largest value is the one after which the performance stops improving. As to the iterative algorithm, three different values for the number of maximum neural units have been considered, and the desired accuracy has been set to 0. Then, three levels for the K candidate weight values at each iteration have been considered, namely, $K = 1$, $K = 10$, and $K = 100$. The experimental setup is summarized in Table I.

To test their use as alternatives to pure random i.i.d. sampling with uniform distribution, denoted as UIID in the following, two different algorithms to generate LDSs in the form of (t, n)-sequences have been considered, specifically the Sobol' sequence and the Niederreiter–Xing sequence (see [11] for details). The two LDSs are denoted in the following as LD-SB and LD-NX, respectively. The first one is implemented in the MATLAB Statistics Toolbox, while the MATLAB code for the Niederreiter–Xing sequence has been obtained from http://people.cs.kuleuven.be/~dirk.nuyens/qmc-generators/. For each of the two training algorithms, each kind of sampling design and each value of N and K, 200 different training runs have been performed, to ensure robustness of the results, using 200 different instances of a given sampling method. While this is straightforward for UIID sampling, in the case of LDSs different instances of a given length N have been obtained by simply taking nonoverlapping portions of N points in a sufficiently long sequence.

TABLE III
AVERAGE MEANS AND STANDARD DEVIATIONS (IN 10^{-1} UNITS)

		$K = 10$			$K = 100$		
		N_1	N_2	N_3	N_1	N_2	N_3
ABA	**LD-SB**	$0.0852, 0.0075$	$0.0828, 0.0067$	$0.0800, 0.0059$	$0.0740, 0.0033$	$0.0728, 0.0029$	$0.0698, 0.0022$
	LD-NX	$0.0850, 0.0079$	$0.0827, 0.0073$	$0.0807, 0.0063$	$0.0747, 0.0036$	$0.0723, 0.0032$	$0.0709, 0.0024$
	UIID	$0.0861, 0.0083$	$0.0830, 0.0073$	$0.0808, 0.0060$	$0.0742, 0.0031$	$0.0722, 0.0031$	$0.0706, 0.0021$
AIR	**LD-SB**	$0.2038, 0.0149$	$0.1926, 0.0088$	$0.1820, 0.0059$	$0.1764, 0.0033$	$0.1714, 0.0028$	$0.1676, 0.0028$
	LD-NX	$0.2116, 0.0167$	$0.01990, 0.0116$	$0.1853, 0.0067$	$0.1774, 0.0034$	$0.1723, 0.0028$	$0.1680, 0.0027$
	UIID	$0.2096, 0.0185$	$0.1961, 0.0116$	$0.1850, 0.0063$	$0.1783, 0.0040$	$0.1725, 0.0028$	$0.1684, 0.0024$
CON	**LD-SB**	$0.2250, 0.0209$	$0.1932, 0.0104$	$0.1819, 0.0075$	$0.1892, 0.0090$	$0.1752, 0.0063$	$0.1646, 0.0050$
	LD-NX	$0.2286, 0.0244$	$0.1957, 0.0104$	$0.1834, 0.0091$	$0.1899, 0.0103$	$0.1750, 0.0061$	$0.1640, 0.0044$
	UIID	$0.2302, 0.0241$	$0.1971, 0.0123$	$0.1840, 0.0086$	$0.1907, 0.0099$	$0.1752, 0.0065$	$0.1656, 0.0057$
PRO	**LD-SB**	$0.7151, 0.0146$	$0.6815, 0.0107$	$0.6628, 0.0073$	$0.6734, 0.0099$	$0.6535, 0.0052$	$0.6395, 0.0032$
	LD-NX	$0.7163, 0.0168$	$0.6812, 0.0112$	$0.6629, 0.0081$	$0.6749, 0.0100$	$0.6538, 0.0058$	$0.6394, 0.0028$
	UIID	$0.7190, 0.0189$	$0.6850, 0.0120$	$0.6655, 0.0081$	$0.6745, 0.0103$	$0.6538, 0.0056$	$0.6390, 0.0030$
INV	**LD-SB**	$0.0386, 0.0037$	$0.0367, 0.0038$	$0.0360, 0.0035$	$0.0343, 0.0035$	$0.0314, 0.0031$	$0.0293, 0.0028$
	LD-NX	$0.0382, 0.0030$	$0.00366, 0.0027$	$0.0357, 0.0026$	$0.0341, 0.0035$	$0.0319, 0.0029$	$0.0294, 0.0027$
	UIID	$0.0382, 0.0032$	$0.0370, 0.0033$	$0.0357, 0.0027$	$0.0346, 0.0034$	$0.0323, 0.0029$	$0.0296, 0.0029$
F20	**LD-SB**	$0.1171, 0.0153$	$0.0689, 0.0045$	$0.0638, 0.0042$	$0.0689, 0.0047$	$0.0631, 0.0046$	$0.0600, 0.0040$
	LD-NX	$0.1257, 0.0199$	$0.0712, 0.0062$	$0.0638, 0.0047$	$0.0693, 0.0047$	$0.0626, 0.0042$	$0.0598, 0.0039$
	UIID	$0.1220, 0.0196$	$0.0699, 0.0049$	$0.0635, 0.0047$	$0.0698, 0.0042$	$0.0626, 0.0042$	$0.0606, 0.0039$
MUa	**LD-SB**	$0.4471, 0.0484$	$0.4255, 0.0372$	$0.4095, 0.0299$	$0.4028, 0.01430$	$0.3972, 0.0140$	$0.3889, 0.0158$
	LD-NX	$0.4504, 0.0643$	$0.4285, 0.0467$	$0.4103, 0.0315$	$0.4012, 0.0146$	$0.3958, 0.0146$	$0.3907, 0.0151$
	UIID	$0.4619, 0.1006$	$0.4318, 0.0617$	$0.4129, 0.03660$	$0.4039, 0.0141$	$0.3973, 0.0143$	$0.3907, 0.0287$
MUb	**LD-SB**	$0.0462, 0.0035$	$0.0383, 0.0013$	$0.0347, 0.0012$	$0.3173, 0.0622$	$0.1882, 0.0429$	$0.1058, 0.0210$
	LD-NX	$0.0495, 0.0035$	$0.0409, 0.0022$	$0.0370, 0.0020$	$0.3740, 0.0899$	$0.2572, 0.0656$	$0.1774, 0.0452$
	UIID	$0.0482, 0.0028$	$0.0401, 0.0020$	$0.0366, 0.0018$	$0.3653, 0.0778$	$0.2382, 0.0535$	$0.1393, 0.0295$
SLI	**LD-SB**	$0.3312, 0.0763$	$0.2872, 0.0628$	$0.2612, 0.0446$	$0.2520, 0.0425$	$0.2444, 0.0381$	$0.2387, 0.0349$
	LD-NX	$0.3183, 0.0678$	$0.2879, 0.0561$	$0.2561, 0.0434$	$0.2558, 0.0412$	$0.2523, 0.0411$	$0.2429, 0.0357$
	UIID	$0.3324, 0.0687$	$0.2916, 0.0657$	$0.2649, 0.0447$	$0.2612, 0.0443$	$0.2494, 0.0401$	$0.2405, 0.0353$

The performance of the estimator obtained by a given sampling method for a single run is measured by the mean squared error (MSE) over the points in the test set. Tables II and III summarize the results for all the tests, reporting the average and standard deviation of the MSEs over the 200 runs.

Then, to provide further statistical evidence, p-values corresponding to the test of the hypothesis that the MSE of the error yielded by an LDS is larger than the one provided by random sampling have been computed. Such p-values have been obtained by comparing all the MSEs obtained for the points in the test set in all the 200 runs with both kinds of samplings (e.g., LD-SB versus UIID) through a pairwise t-test with unequal variances. Notice that the smaller is the p-value, the more unlikely is that the mean error given by the low-discrepancy sets is larger than the error yielded by the random ones. This means that the UIID formally loses over the low-discrepancy sampling scheme when the p-value is <0.5, and a value <0.05 can be considered as a highly statistically significant proof that the low-discrepancy set is expected to yield lower errors systematically.

To wrap up the results of the tests, Table IV reports some significant statistics. In particular, the table reports as follows.

1) The number of times, over all the test problems, that a specific kind of sampling yields the best results (minimum average MSE over the 200 runs, denoted as $\overline{\text{MSE}}$ in the table).
2) The number of times a specific low-discrepancy scheme yields a lower $\overline{\text{MSE}}$ with respect to UIID sampling.
3) The number of times the UIID sampling loses, in terms of p-value, against a specific low-discrepancy scheme, i.e., the p-value is >0.5, highlighting the times the p-value is <0.05.

TABLE IV
SUMMARY OF THE OBTAINED RESULTS

times best $\overline{\text{MSE}}$ given by:
LD-SB: 74 (68.5%)
LD-NX: 29 (26.9%)
UIID: 5 (4.6%)
times LD-SB lower $\overline{\text{MSE}}$ than UIID: 94 (87.0%)
times LD-NX lower $\overline{\text{MSE}}$ than UIID: 75 (69.4%)
UIID losses vs LD-SB w.r.t. p-value: 94 (87.0%); 67 times $p < .05$
UIID losses vs LD-NX w.r.t. p-value: 75 (69.4%); 42 times $p < .05$

The results confirm in practice that a deterministic assignment of the hidden weight values can be advantageous when sequences aimed at minimizing the discrepancy are employed. The tests show a quite evident superiority of LDSs over uniform random sampling with all the data sets in all the conditions, while the UIID sets yielded the best outcomes only 4.6% of the times. Concerning the low-discrepancy sets, looking at the tables, it turns out that Sobol' sequences provide the best results overall. This, coupled with the fact that they are the ones more frequently implemented in software packages (here, as said, the MATLAB Statistics Toolbox version was employed), indicates them as very promising sequences to be employed in ELM applications.

REFERENCES

[1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[2] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3460–3468, Oct. 2008.
[3] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: A survey," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 2, pp. 107–122, Jun. 2011.
[4] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.
[5] Z. Yan and J. Wang, "Robust model predictive control of nonlinear systems with unmodeled dynamics and bounded uncertainties based on neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 457–469, Mar. 2014.
[6] J. Luo, C.-M. Vong, and P.-K. Wong, "Sparse Bayesian extreme learning machine for multi-classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 4, pp. 836–843, Apr. 2014.
[7] E. Cambria *et al.*, "Extreme learning machines," *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 30–59, Nov./Dec. 2013.
[8] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.
[9] A. Castaño, F. Fernández-Navarro, and C. Hervás-Martínez, "PCA-ELM: A robust and pruned extreme learning machine approach based on principal component analysis," *Neural Process. Lett.*, vol. 37, no. 3, pp. 377–392, Jun. 2013.
[10] K.-T. Fang and Y. Wang, *Number-Theoretic Methods in Statistics*. London, U.K.: Chapman & Hall, 1994.
[11] J. Dick and F. Pillichshammer, *Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration*. New York, NY, USA: Cambridge Univ. Press, 2010.
[12] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia, PA, USA: SIAM, 1992.
[13] C. Cervellera and D. Macciò, "Local linear regression for function learning: An analysis based on sample discrepancy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 11, pp. 2086–2098, Nov. 2014.
[14] C. Cervellera, M. Gaggero, and D. Macciò, "Low-discrepancy sampling for approximate dynamic programming with local approximators," *Comput. Oper. Res.*, vol. 43, pp. 108–115, Mar. 2014.
[15] C. Cervellera and D. Macciò, "Learning with kernel smoothing models and low-discrepancy sampling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 504–509, Mar. 2013.
[16] K.-L. Chung, "An estimate concerning the Kolmogoroff limit distribution," *Trans. Amer. Math. Soc.*, vol. 67, no. 1, pp. 36–50, Sep. 1949.
[17] G. B. Folland, *Real Analysis: Modern Techniques and Their Applications*, 2nd ed. New York, NY, USA: Wiley, 1999.
[18] M. Stinchcombe and H. White, "Approximating and learning unknown mappings using multilayer feedforward networks with bounded weights," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 3. San Diego, CA, USA, Jun. 1990, pp. 7–16.
[19] K. Bache and M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml
[20] C. Cervellera and D. Macciò, "A comparison of global and semi-local approximation in T-stage stochastic optimization," *Eur. J. Oper. Res.*, vol. 208, no. 2, pp. 109–118, Jan. 2011.