Neurocomputing ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Neurocomputing

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

An oscillation bound of the generalization performance of extreme learning machine and corresponding analysis

Di Wang [a,*], Ping Wang [a], Yan Ji [b]

[a] Department of Electrical Engineering and Automation, Tianjin University, 300072 Tianjin, China
[b] Tianjin Branch, China Merchants Bank, 300201 Tianjin, China

ARTICLE INFO

Article history:
Received 8 June 2014
Received in revised form
28 August 2014
Accepted 2 October 2014
Communicated by G.-B. Huang

Keywords:
Extreme learning machine
Oscillation bound
Generalization performance
Theoretical research
Infinite hidden nodes

ABSTRACT

Extreme Learning Machine (ELM), proposed by Huang et al. in 2004 for the first time, performs better than traditional learning machines such as BP networks and SVM in some applications. This paper attempts to give an oscillation bound of the generalization performance of ELM and a reason why ELM is not sensitive to the number of hidden nodes, which are essential open problems proposed by Huang et al. in 2011. The derivation of the bound is in the framework of statistical learning theory and under the assumption that the expectation of the ELM kernel exists. It turns out that our bound is consistent with the experimental results about ELM obtained before and predicts that overfitting can be avoided even when the number of hidden nodes approaches infinity. The prediction is confirmed by our experiments on 15 data sets using one kind of activation function with every parameter independently drawn from the same Guasssian distribution, which satisfies the assumption above. The experiments also showed that when the number of hidden nodes approaches infinity, the ELM kernel with the activation is insensitive to the kernel parameter.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

To make the single-layer feedforward neural networks (SLFN) learn faster, a novel learning method called Extreme Learning Machine (ELM) was first proposed by Huang et al. in 2004 [1]. ELM owns the features that the parameters of hidden nodes can be randomly assigned and only the output weights need to be determined analytically. It can give better generalization performance with less learning time and human intervene than those traditional techniques such as BP networks and SVM [2–5]. Owing to its effectiveness and efficiency in many applications, ELM has attracted much attention from various fields in recent years [6–13].

Although many researches concerning theory and application study of ELM have been conducted [14–20], some open problems need to be solved as soon as possible [2]. This paper makes an attempt to answer the first two questions proposed in [2] that is why ELM is not sensitive to the number of hidden nodes and how to estimate the oscillation bound of the generalization performance of ELM.

To our best knowledge, this is the first paper which attempts to answer the questions above directly. Our work is mainly based on the theoretical analysis about ELM that ELM is a kind of kernel method [2,3,21–24] and on the experimental observation that the performance of the ELM kernel with randomly generated hidden nodes approaching an infinite network kernel [25,26]. The details are given in the next section.

The rest of this paper is organized as follows. Section 2 briefly reviews ELM. In Section 3, the derivation of the bound and corresponding analysis are given. Experiments are introduced in Section 4 and eventually, in Section 5, the conclusions of this paper are drawn.

2. Extreme learning machine

This paper only covers the situation that the network has one output node and it is easy to extend the analysis to multi-output node cases.

The output function of SLFN with one output node is

$$f(\mathbf{x}) = \sum_{j=1}^{L} \beta_j G(\mathbf{x}, \mathbf{w}_j, b_j)$$

where $\mathbf{x} \in \mathbf{R}^n$ is the input vector, L is the number of hidden nodes, $G(\mathbf{x}, \mathbf{w}_j, b_j)$ is the jth hidden node activation function with the parameters $(\mathbf{w}_j, b_j) \in \mathbf{R}^n \times \mathbf{R}$ and $\beta_j \in \mathbf{R}$ is the weight between the jth hidden node and the output node. For additive nodes with activation function g, $G(\mathbf{x}, \mathbf{w}_j, b_j) = g(\mathbf{w}_j^T \mathbf{x} + b_j)$, and for RBF nodes, $G(\mathbf{x}, \mathbf{w}_j, b_j) = g(\|\mathbf{x} - \mathbf{w}_j\| / b_j)$ [27]. It has been shown in [28,29] that

* Corresponding author.
E-mail address: wangditju2012@126.com (D. Wang).

http://dx.doi.org/10.1016/j.neucom.2014.10.006
0925-2312/© 2014 Elsevier B.V. All rights reserved.

if $G(\mathbf{x}, \mathbf{w}_j, b_j)$ satisfies some mild conditions and all the parameters containing L, β_j and $(\mathbf{w}_j, b_j)$ can be freely adjusted, SLFN can approximate any continuous function at any degree of accuracy. However, from practice point of view, tuning all the parameters requires a large amount of time and human intervene, and thus it limits the applications of the traditional SLFN. To make SLFN learn faster, a new theory is needed to guide practice.

It is Huang et al. who have proved that as long as L is large enough, we can assign $(\mathbf{w}_j, b_j)$ randomly and determine the β_j analytically to approximate the target function as accurately as possible [27,30]. This theory gives rise to a fast learning algorithm which is called Extreme Learning Machine [2].

For a hidden node number L, activation function $G(\mathbf{x}, \mathbf{w}_j, b_j)$ and a training set $N = \{(\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathbf{R}^n, y_i \in \mathbf{R},\ i = 1, 2, \ldots, l\}$, where $\mathbf{x}_i$ is the input and y_i is the target output, ELM randomly generates L hidden node parameters $\{(\mathbf{w}_j, b_j), \mathbf{w}_j \in \mathbf{R}^n, b_j \in \mathbf{R},\ j = 1, 2, \ldots, L\}$ and then calculates the hidden layer output matrix $\mathbf{H}$, where

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \mathbf{h}(\mathbf{x}_2) \\ \vdots \\ \mathbf{h}(\mathbf{x}_l) \end{bmatrix} = \begin{bmatrix} G(\mathbf{x}_1, \mathbf{w}_1, b_1) & \cdots & G(\mathbf{x}_1, \mathbf{w}_L, b_L) \\ G(\mathbf{x}_2, \mathbf{w}_1, b_1) & \cdots & G(\mathbf{x}_2, \mathbf{w}_L, b_L) \\ \vdots & \vdots & \vdots \\ G(\mathbf{x}_l, \mathbf{w}_1, b_1) & \cdots & G(\mathbf{x}_l, \mathbf{w}_L, b_L) \end{bmatrix}$$

At last, it calculates the output weight vector

$$\boldsymbol{\beta}^\star = [\beta_1^\star, \beta_2^\star, \ldots, \beta_L^\star]^T = \mathbf{H}^+ \mathbf{Y}$$

where $\mathbf{H}^+$ is the Moore–Penrose pseudo-inverse of $\mathbf{H}$ and $\mathbf{Y} = [y_1, y_2, \ldots, y_l]^T$. The output function given by ELM is $f(\mathbf{x}) = \sum_{j=1}^{L} \beta_j^\star G(\mathbf{x}, \mathbf{w}_j, b_j)$.

It can be seen that after hidden node parameters being randomly assigned, ELM aims to output a function whose $\boldsymbol{\beta}^\star$ is the minimum norm least-squares solution of $\mathbf{H}\boldsymbol{\beta} = \mathbf{Y}$. In other words, for a fixed L and L parameters $(\mathbf{w}_j, b_j)$, ELM is based on empirical risk minimization principle and its empirical risk is $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|$ ($\|\cdot\|$ is the 2-norm of a vector).

To make the resultant solution more stable and have better generalization performance, Huang et al. suggested that according to ridge regression theory which balances the empirical risk and the complexity of the network, the output function given by ELM can be

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x})\mathbf{H}^T \left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T\right)^{-1} \mathbf{Y}$$

or

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}^T\mathbf{H}\right)^{-1} \mathbf{H}^T\mathbf{Y}$$

where $\mathbf{I}$ is the identity matrix with proper size and λ is a positive constant [2,3]. This is equivalent to minimizing

$$\lambda\|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|^2 + \|\boldsymbol{\beta}\|^2$$

or minimizing

$$\|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|^2 \quad \text{subject to } \|\boldsymbol{\beta}\|^2 \le c$$

where c is a positive constant related to λ [31].

Some researchers, such as Liu et al. [21], Frénay et al. [22,23], Huang et al. [2,3,24] and Parviainen et al. [25,26], pointed out that ELM can be viewed as a kernel method because for a fixed L and L parameters $(\mathbf{w}_j, b_j)$, ELM first maps the input to a L-dimensional feature space, where each dimension corresponds to a hidden node, and then finds a linear relation between the feature space vector and the corresponding output. Frénay et al. [22,23] defined ELM kernel function as $\kappa_{ELM}(\mathbf{x}_i, \mathbf{x}_j) = (1/L)\mathbf{h}(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_j)^T$, and used it successfully for SVM and SVR. Huang et al. [2,3,24] also realized that the output function of ELM can be written compactly as

$$f(\mathbf{x}) = [\mathbf{h}(\mathbf{x})\mathbf{h}(\mathbf{x}_1)^T, \mathbf{h}(\mathbf{x})\mathbf{h}(\mathbf{x}_2)^T, \ldots, \mathbf{h}(\mathbf{x})\mathbf{h}(\mathbf{x}_l)^T] \left(\frac{\mathbf{I}}{\lambda} + \boldsymbol{\Omega}_{l\times l}\right)^{-1} \mathbf{Y},$$

where $\boldsymbol{\Omega}_{l\times l} = [\mathbf{h}(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_j)^T]_{l\times l}$. The work of Parviainen et al. [25,26] showed that the ELM kernel $\kappa_{ELM}(\mathbf{x}_i, \mathbf{x}_j) = (1/L)\mathbf{h}(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_j)^T$ approximates an infinite network kernel as L grows. They interpreted ELM as an approximation to a network with infinite number of hidden nodes.

All the above-mentioned works have inspired our research work. Firstly, if we fix L and L parameters $(\mathbf{w}_j, b_j)$, ELM has a generalization bound based on the kernel method. Secondly, the performance of the ELM kernel with randomly generated hidden nodes approaching an infinite network kernel implies that there is the Law of Large Numbers between the ELM kernel and the corresponding infinite network kernel and we can bound the ELM kernel using a probability inequality properly. Combining the two bounds above, we achieve an oscillation bound of the generalization performance of ELM.

3. The solutions of the two open problems of ELM

In this section, we first give an oscillation bound of the generalization performance of ELM, and then give some analysis about it.

3.1. An oscillation bound of the generalization performance of ELM

Our results are obtained in the framework of statistical learning theory [32].

With reference to [33, 16.1 introduction], we assume that the data $(\mathbf{X}, Y)$ is drawn according to a probability distribution $\mu_{\mathbf{Z}}$ on $\mathbf{R}^n \times \mathbf{R}$ where the support of Y is $[-D, D]$ (D is a positive constant) and we use the expected quadratic loss, $E_{\mu_{\mathbf{Z}}}[(f(\mathbf{X}) - Y)^2]$, to measure how accurately $f(\mathbf{X})$ approximates Y. We also assume that the function given by ELM is bounded.

3.1.1. Notations

Here, the meanings of the following notations in the rest of this paper are given.

$\boldsymbol{\Theta}^L$ denotes L *random parameters* $\{(\mathbf{W}_j, B_j), j = 1, 2, \ldots, L\}$ drawn independently according to a probability distribution $\mu_{\boldsymbol{\Theta}}$ on $\mathbf{R}^n \times \mathbf{R}$;

θ^L denotes L *parameters* $\{(\mathbf{w}_j, b_j), \mathbf{w}_j \in \mathbf{R}^n, b_j \in \mathbf{R}, j = 1, 2, \ldots, L\}$;

$\boldsymbol{Z}^l$ denotes l *random data* $\{(\mathbf{X}_i, Y_i),\ i = 1, 2, \ldots, l\}$ drawn independently according to a probability distribution $\mu_{\mathbf{Z}}$ on $\mathbf{R}^n \times \mathbf{R}$, where the support of Y is $[-D, D]$;

$\mathbf{z}^l$ denotes l *data* (training set) $\{(\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathbf{R}^n, y_i \in [-D, D], i = 1, 2, \ldots, l\}$.

$G(\mathbf{x}, \mathbf{w}_j, b_j)$ is a kind of activation function parameterized by $(\mathbf{w}_j, b_j)$ and its range is included in $[-1, 1]$;

$\boldsymbol{\phi}_{\boldsymbol{\Theta}^L}(\mathbf{x}) = [G(\mathbf{x}, \mathbf{W}_1, B_1), G(\mathbf{x}, \mathbf{W}_2, B_2) \cdots G(\mathbf{x}, \mathbf{W}_L, B_L)]^T$ is a random vector for a fixed $\mathbf{x}$;

$\boldsymbol{\phi}_{\theta^L}(\mathbf{x}) = [G(\mathbf{x}, \mathbf{w}_1, b_1), G(\mathbf{x}, \mathbf{w}_2, b_2) \cdots G(\mathbf{x}, \mathbf{w}_L, b_L)]^T$.

The L-dimensional feature space mentioned above is denoted by $\boldsymbol{\Psi}_L$ and its inner product is the L-dimensional Euclidean inner product divided by L. $\langle\cdot,\cdot\rangle_L$ denotes the inner product of $\boldsymbol{\Psi}_L$. For example, for $\boldsymbol{\phi}_{\theta^L}(\mathbf{x}_i), \boldsymbol{\phi}_{\theta^L}(\mathbf{x}_j) \in \boldsymbol{\Psi}_L$, $\langle \boldsymbol{\phi}_{\theta^L}(\mathbf{x}_i), \boldsymbol{\phi}_{\theta^L}(\mathbf{x}_j)\rangle_L = (1/L)\boldsymbol{\phi}_{\theta^L}(\mathbf{x}_i)^T \boldsymbol{\phi}_{\theta^L}(\mathbf{x}_j)$.

For fixed θ^L, the output function space of ELM with domain $\mathbf{R}^n$ is $F_C(\theta^L) = \{\mathbf{x} \mapsto \langle \boldsymbol{\phi}_{\theta^L}(\mathbf{x}), \boldsymbol{\beta}\rangle_L | \boldsymbol{\beta} \in \boldsymbol{\Psi}_L, \|\boldsymbol{\beta}\|_L \le C\}$, where $\|\cdot\|_L$ is the induced norm of $\langle\cdot,\cdot\rangle_L$ and C is a positive constant to bound the output function given by ELM.

As $|G(\mathbf{x}, \mathbf{w}_j, b_j)| \le 1$ implies $\|\boldsymbol{\phi}_{\theta^L}(\mathbf{x})\|_L \le 1$, $|\langle \boldsymbol{\phi}_{\theta^L}(\mathbf{x}), \boldsymbol{\beta}\rangle_L| \le \|\boldsymbol{\phi}_{\theta^L}\|_L \|\boldsymbol{\beta}\|_L \le C$, i.e. the function given by ELM is bounded by C.

$E_{\mu_{\mathbf{Z}}}(f)$ denotes $E_{\mu_{\mathbf{Z}}}[(f(\mathbf{X}) - Y)^2]$.

Please cite this article as: D. Wang, et al., An oscillation bound of the generalization performance of extreme learning machine and corresponding analysis, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.10.006

$\hat{E}_{\mathbf{z}^l}(f)$ denotes $(1/l)\sum_{i=1}^{l}(f(\mathbf{x}_i)-y_i)^2$ and $\hat{E}_{\mathbf{Z}^l}(f)$ denotes $(1/l)\sum_{i=1}^{l}(f(\mathbf{X}_i)-Y_i)^2$. For a fixed function f, $\hat{E}_{\mathbf{z}^l}(f)$ is a *valuable* depending on $\mathbf{z}^l$ and $\hat{E}_{\mathbf{Z}^l}(f)$ is a *random valuable* depending on $\mathbf{Z}^l$.

$$\kappa_{\theta^L}(\mathbf{x}_i,\mathbf{x}_j)=\frac{1}{L}\boldsymbol{\phi}_{\theta^L}(\mathbf{x}_i)^T\boldsymbol{\phi}_{\theta^L}(\mathbf{x}_j),$$

$$\kappa_{\boldsymbol{\Theta}^L}(\mathbf{x}_i,\mathbf{x}_j)=\frac{1}{L}\boldsymbol{\phi}_{\boldsymbol{\Theta}^L}(\mathbf{x}_i)^T\boldsymbol{\phi}_{\boldsymbol{\Theta}^L}(\mathbf{x}_j),$$

$$\kappa(\mathbf{x}_i,\mathbf{x}_j)=E_{\mu_{\boldsymbol{\Theta}}}[G(\mathbf{x}_i,\mathbf{W},B),G(\mathbf{x}_j,\mathbf{W},B)],$$

$$\hat{\kappa}_{\theta^L}(\mathbf{x}^l)=\frac{1}{l}\sum_{i=1}^{l}\kappa_{\theta^L}(\mathbf{x}_i,\mathbf{x}_i),$$

$$\hat{\kappa}_{\theta^L}(\mathbf{X}^l)=\frac{1}{l}\sum_{i=1}^{l}\kappa_{\theta^L}(\mathbf{X}_i,\mathbf{X}_i),$$

$$\hat{\kappa}_{\boldsymbol{\Theta}^L}(\mathbf{x}^l)=\frac{1}{l}\sum_{i=1}^{l}\kappa_{\boldsymbol{\Theta}^L}(\mathbf{x}_i,\mathbf{x}_i),$$

$$\hat{\kappa}(\mathbf{x}^l)=\frac{1}{l}\sum_{i=1}^{l}\kappa(\mathbf{x}_i,\mathbf{x}_i),$$

$$\hat{\kappa}(\mathbf{X}^l)=\frac{1}{l}\sum_{i=1}^{l}\kappa(\mathbf{X}_i,\mathbf{X}_i),$$

$$Q=D+C,$$

$P\{\star\}$ denotes the probability of the random event "$\star$".

3.1.2. Lemmas

To estimate the generalization bound of a learning machine, we need a method to measure the capacity of a function class. In this paper, we use the empirical Rademacher complexity to measure the capacity [31,34]. Empirical Rademacher complexity is closely related to kernel and that is exactly what we need.

Definition 1 (*Shawe-Taylor and Cristianini [31, Defination 4.8]*). For $\{\mathbf{M}_i,\mathbf{M}_i\in\mathcal{M},i=1,2,\ldots,l\}$ generated by a distribution $\mu_{\mathbf{M}}$ on space $\mathcal{M}$, the empirical Rademacher complexity of a real-valued function class F with domain $\mathcal{M}$ is the random variable

$$\hat{R}_l(F)=E_\rho\left[\sup_{f\in F}\left|\frac{2}{l}\sum_{i=1}^{l}\rho_i f(\mathbf{M}_i)\right|\middle|\mathbf{M}_1,\mathbf{M}_2,\ldots,\mathbf{M}_l\right]$$

where $\rho=\{\rho_i,i=1,2,\ldots,l\}$ are random variables which take values -1 and $+1$ independently with equal probability 0.5.

Lemma 1 (*Shawe-Taylor and Cristianini [31, Theorem 4.15(vi)]*). *Let $\{\mathbf{M}_i,\mathbf{M}_i\in\mathcal{M},i=1,2,\ldots,l\}$ be generated by a distribution $\mu_{\mathbf{M}}$ on space $\mathcal{M}$. For any $1\le q\le\infty$, let $L_{F,h,q}=\{|f-h|^q|f\in F\}$, where F is a real-valued function class with domain $\mathcal{M}$, and h is a function with the same domain. If $\|f-h\|_\infty\le 1$ for every $f\in F$, then*

$$\hat{R}_l(L_{F,h,q})\le 2q\left(\hat{R}_l(F)+2\sqrt{\sum_{i=1}^{l}\frac{h(\mathbf{M}_i)^2}{l^2}}\right).$$

Lemma 2 (*Shawe-Taylor and Cristianini [31, Theorem 4.9]*). *Fix $\delta\in(0,1)$ and let F be a class of functions mapping from $\mathbf{R}^n\times\mathbf{R}$ to $[0,1]$. For $\mathbf{Z}^l$ mentioned above and every $f\in F$, the following inequality holds:*

$$P\left\{E_{\mu_{\mathbf{Z}}}[f(\mathbf{X},Y)]\le\frac{1}{l}\sum_{i=1}^{l}f(\mathbf{X}_i,Y_i)+\hat{R}_l(F)+3\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2l}}\right\}\ge 1-\delta.$$

Lemma 3 (*[31, Theorem 4.12]*). *Let $\{\mathbf{X}_i,i=1,2\ldots,l\}$ be drawn independently according to a probability distribution $\mu_{\mathbf{X}}$ on $\mathbf{R}^n$, then*

$$\hat{R}_l(F_C(\theta^L))\le\frac{2C}{\sqrt{l}}\sqrt{\hat{\kappa}_{\theta^L}(\mathbf{X}^l)}.$$

Lemma 4 (*Shawe-Taylor and Cristianini [31, pp. 320–321]*). *Suppose $(\mathbf{W},B)\sim\mu_{\boldsymbol{\Theta}}$, then for fixed $\mathbf{x}_i,\mathbf{x}_j\in\mathbf{R}^n$, the following inequality holds when $\kappa(\mathbf{x}_i,\mathbf{x}_j)$ exists:*

$$P\{\kappa_{\boldsymbol{\Theta}^L}(\mathbf{x}_i,\mathbf{x}_j)-\kappa(\mathbf{x}_i,\mathbf{x}_j)\ge\varepsilon\}\le\exp\left(\frac{-L\varepsilon^2}{2}\right).$$

As the expectation of $\kappa_{\boldsymbol{\Theta}^L}(\mathbf{x}_i,\mathbf{x}_j)$ is $\kappa(\mathbf{x}_i,\mathbf{x}_j)$, Lemma 4 is always correct.

Section 3 of [35] gives a simple review of $\kappa(\mathbf{x}_i,\mathbf{x}_j)$ which is derived for infinite neural networks. Especially, for $G(\mathbf{x},\mathbf{W},B)=g(\mathbf{W}^T\mathbf{x}+B)$, where $g(t)=(2/\sqrt{\pi})\int_0^t e^{-z^2}\,dz$ for $t\in\mathbf{R}$, and each component of $\mathbf{W}$ and B obey Gaussian distribution with mean 0 and variance σ^2, we have

$$\kappa(\mathbf{x}_i,\mathbf{x}_j)=\frac{2}{\pi}\sin^{-1}\left(\frac{2[1,\mathbf{x}_i^T]\boldsymbol{\Sigma}[1,\mathbf{x}_j^T]^T}{\sqrt{(1+2[1,\mathbf{x}_i^T]\boldsymbol{\Sigma}[1,\mathbf{x}_i^T]^T)(1+2[1,\mathbf{x}_j^T]\boldsymbol{\Sigma}[1,\mathbf{x}_j^T]^T)}}\right)\tag{1}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix and every element of the diagonal is σ^2.

Eq. (1) is also mentioned in [23,25] for the analysis and the experiments about ELM.

3.1.3. Theorems

First we give the generalization bound for fixed θ^L.

Theorem 1. *For $\mathbf{Z}^l$ and fixed θ^L mentioned above and every $f\in F_C(\theta^L)$, the following inequality holds:*

$$P\left\{E_{\mu_{\mathbf{Z}}}(f)\le\hat{E}_{\mathbf{Z}^l}(f)+\frac{8Q}{\sqrt{l}}(D+C\sqrt{\hat{\kappa}_{\theta^L}(\mathbf{X}^l)})+3Q^2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2l}}\right\}\ge 1-\delta$$

where $\delta\in(0,1)$.

Proof. For every $f\in F_C(\theta^L)$ and $y\in[-D,D]$, we can write

$$(f(\mathbf{x})-y)^2\le(D+C)^2=Q^2.$$

Let $K=\{(\mathbf{x},y)\mapsto(f(\mathbf{x})-y)^2/Q^2|f(\mathbf{x})\in F_C(\theta^L)\}$, $K_1=\{(\mathbf{x},y)\mapsto f(\mathbf{x})/Q|f(\mathbf{x})\in F_C(\theta^L)\}$, and $h(\mathbf{x},y)=y/Q$, then $K=\{|k_1-h|^2|k_1\in K_1\}$.

By Lemma 1 and $|Y_i|\le D$,

$$\begin{aligned}\hat{R}_l(K)&\le 4\left(\hat{R}_l(K_1)+\frac{2\sqrt{\sum_{i=1}^{l}Y_i^2}}{Ql}\right)\\&\le 4\left(\hat{R}_l(K_1)+\frac{2D}{Q\sqrt{l}}\right).\end{aligned}$$

By Definition 1 and Lemma 3,

$$\begin{aligned}\hat{R}_l(K_1)&=E_\rho\left[\sup_{f\in F}\left|\frac{2}{l}\sum_{i=1}^{l}\frac{\rho_i f(\mathbf{X}_i)}{Q}\right|\middle|\mathbf{Z}^l\right]\\&=E_\rho\left[\sup_{f\in F}\left|\frac{2}{l}\sum_{i=1}^{l}\frac{\rho_i f(\mathbf{X}_i)}{Q}\right|\middle|\mathbf{X}_1,\mathbf{X}_2,\ldots,\mathbf{X}_l\right]\\&\le\frac{2C}{Q\sqrt{l}}\sqrt{\hat{\kappa}_{\theta^L}(\mathbf{X}^l)},\end{aligned}$$

so

$$\hat{R}_l(K)\le 4\left(\frac{2C}{Q\sqrt{l}}\sqrt{\hat{\kappa}_{\theta^L}(\mathbf{X}^l)}+\frac{2D}{Q\sqrt{l}}\right)\tag{2}$$

Please cite this article as: D. Wang, et al., An oscillation bound of the generalization performance of extreme learning machine and corresponding analysis, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.10.006

By Lemma 2, for every $(f(\mathbf{x})-y)^2/Q^2 \in K$, with probability at least $1-\delta$, the following inequality holds:

$$E_{\mu_{\mathbf{Z}}}\left[\frac{(f(\mathbf{X})-Y)^2}{Q^2}\right] \le \frac{1}{l}\sum_{i=1}^{l}\frac{(f(\mathbf{X}_i)-Y_i)^2}{Q^2}+\hat{R}_l(K)+3\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2l}}$$

which is equivalent to

$$E_{\mu_{\mathbf{Z}}}(f) \le \hat{E}_{\mathbf{Z}^l}(f)+Q^2\hat{R}_l(K)+3Q^2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2l}}. \tag{3}$$

By substituting (2) into (3), we complete the proof. □

Our belief that why ELM performs well can be found in the Law of Large Numbers guides us to combine the results of Lemma 4 and Theorem 1, through which the oscillation bound of the generalization performance of ELM can be given.

Theorem 2. *Suppose that for every $\mathbf{x} \in \mathbf{R}^n$, $\kappa(\mathbf{x},\mathbf{x})$ exists and θ^L, $\mathbf{z}^l$ are realizations of $\mathbf{\Theta}^L$, $\mathbf{Z}^l$ respectively. If we first get θ^L and z^l, and second choose any $f \in F_C(\theta^L)$, then the following inequality holds:*

$$P\left\{E_{\mu_{\mathbf{Z}}}(f) \le \hat{E}_{\mathbf{Z}^l}(f)+\frac{8Q}{\sqrt{l}}\left(D+C\sqrt{\hat{\kappa}(\mathbf{X}^l)+\sqrt{\frac{2}{L}\ln\left(\frac{l}{\delta^\star}\right)}}\right) +3Q^2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2l}}\right\} \ge 1-\delta-\delta^\star$$

where δ, $\delta^\star$ and $(\delta+\delta^\star) \in (0,1)$.

Proof. The randomness is from both $\mathbf{\Theta}^L$ and $\mathbf{Z}^l$.

For fixed $\mathbf{z}^l$ and by Lemma 4

$$\begin{aligned}
P\left\{\hat{\kappa}_{\mathbf{\Theta}^L}(\mathbf{x}^l)-\hat{\kappa}(\mathbf{x}^l) \le \varepsilon\right\} &= P\left\{\sum_{i=1}^{l}\kappa_{\mathbf{\Theta}^L}(\mathbf{x}_i,\mathbf{x}_i)-\sum_{i=1}^{l}\kappa(\mathbf{x}_i,\mathbf{x}_i) \le l\varepsilon\right\}\\
&\ge P\left\{\bigcap_{i=1}^{l}\{\kappa_{\mathbf{\Theta}^L}(\mathbf{x}_i,\mathbf{x}_i)-\kappa(\mathbf{x}_i,\mathbf{x}_i) \le \varepsilon\}\right\}\\
&= 1-P\left\{\bigcup_{i=1}^{l}\{\kappa_{\mathbf{\Theta}^L}(\mathbf{x}_i,\mathbf{x}_i)-\kappa(\mathbf{x}_i,\mathbf{x}_i) > \varepsilon\}\right\}\\
&\ge 1-\sum_{i=1}^{l}P\{\kappa_{\mathbf{\Theta}^L}(\mathbf{x}_i,\mathbf{x}_i)-\kappa(\mathbf{x}_i,\mathbf{x}_i) > \varepsilon\}\\
&\ge 1-l\exp\left(\frac{-L\varepsilon^2}{2}\right)
\end{aligned}$$

Let $l\exp(-L\varepsilon^2/2)=\delta^\star$, then the following inequality holds:

$$P\left\{\hat{\kappa}_{\mathbf{\Theta}^L}(\mathbf{x}^l) \le \hat{\kappa}(\mathbf{x}^l)+\sqrt{\frac{2}{L}\ln\left(\frac{l}{\delta^\star}\right)}\right\} \ge 1-\delta^\star \tag{4}$$

Let $p_{\mathbf{\Theta}^L}(\theta^L)$ denote the probability density function of $\mathbf{\Theta}^L$ and $p_{\mathbf{Z}^l}(\mathbf{z}^l)$ denote the probability density function of $\mathbf{Z}^l$.

Here are some notations for fixed θ^L, $\mathbf{z}^l$ and function f:

$$E_{\mu_{\mathbf{Z}}}(f) \le \hat{E}_{\mathbf{Z}^l}(f)+\frac{8Q}{\sqrt{l}}\left(D+C\sqrt{\hat{\kappa}(\mathbf{x}^l)+\sqrt{\frac{2}{L}\ln\left(\frac{l}{\delta^\star}\right)}}\right)+3Q^2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2l}} \tag{5}$$

$$E_{\mu_{\mathbf{Z}}}(f) \le \hat{E}_{\mathbf{Z}^l}(f)+\frac{8Q}{\sqrt{l}}\left(D+C\sqrt{\hat{\kappa}_{\theta^L}(\mathbf{x}^l)}\right)+3Q^2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2l}} \tag{6}$$

$$\hat{\kappa}_{\theta^L}(\mathbf{x}^l) \le \hat{\kappa}(\mathbf{x}^l)+\sqrt{\frac{2}{L}\ln\left(\frac{l}{\delta^\star}\right)} \tag{7}$$

$M=\{(\theta^L,\mathbf{z}^l)|\ \forall f\ \in F_C(\theta^L),\ f,\ \theta^L \text{ and } \mathbf{z}^l \text{ satisfy (5)}\}$,
$S=\{(\theta^L,\mathbf{z}^l)|\ \forall f\ \in F_C(\theta^L),\ f,\ \theta^L \text{ and } \mathbf{z}^l \text{ satisfy (6)}\}$,
$T=\{(\theta^L,\mathbf{z}^l)|\ \theta^L \text{ and } \mathbf{z}^l \text{ satisfy (7)}\}$.

Then (4) is equivalent to $P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in T|\mathbf{Z}^l=\mathbf{z}^l\} \ge 1-\delta^\star$ and the conclusion of Theorem 1 is equivalent to $P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in S|\mathbf{\Theta}^L=\theta^l\} \ge 1-\delta$.

As

$$\begin{aligned}
P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in M\} &\ge P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in S \cap T\}\\
&= 1-P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in S^c \cup T^c\}\\
&\ge 1-P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in S^c\}-P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in T^c\}\\
&= P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in S\}+P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in T\}-1,
\end{aligned}$$

$$\begin{aligned}
P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in S\} &= \int_{\mathbf{R}^n\times\mathbf{R}} P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in S|\mathbf{\Theta}^L=\theta^L\}\ p_{\mathbf{\Theta}^L}(\theta^L)\,d\theta^L\\
&\ge (1-\delta)\int_{\mathbf{R}^n\times\mathbf{R}} p_{\mathbf{\Theta}^L}(\theta^L)\,d\theta^L\\
&= 1-\delta,
\end{aligned}$$

and

$$\begin{aligned}
P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in T\} &= \int_{\mathbf{R}^n\times[-D,D]} P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in T|\mathbf{Z}^l=\mathbf{z}^l\}\ p_{\mathbf{Z}^l}(\mathbf{z}^l)\,d\mathbf{z}^l\\
&\ge 1-\delta^\star,
\end{aligned}$$

we can conclude that $P\{(\mathbf{\Theta}^L,\mathbf{Z}^l) \in M\} \ge 1-\delta-\delta^\star$ and this completes the proof. □

The oscillation bound of the generalization performance of ELM is given by (5).

The conclusion of Theorem 2 is that for $100(1-\delta-\delta^\star)$ percent of $(\theta^L,\mathbf{z}^l)$, the function given by ELM satisfies (5).

As $\hat{\kappa}(\mathbf{x}^l) \le 1$, the risk of ELM is also bounded by

$$\hat{E}_{\mathbf{Z}^l}(f)+\frac{8Q}{\sqrt{l}}\left(D+C\sqrt{1+\sqrt{\frac{2}{L}\ln\left(\frac{l}{\delta^\star}\right)}}\right)+3Q^2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2l}}. \tag{8}$$

For fixed L and l, ELM balances between the empirical risk $\hat{E}_{\mathbf{Z}^l}(f)$ and the norm of output weights related to C by choosing λ properly, which is a good way to give a fine performance. The ratio of the balance is related to $\sqrt{\hat{\kappa}(\mathbf{x}^l)+\sqrt{(2/L)\ln(l/\delta^\star)}}$ or $\sqrt{1+\sqrt{(2/L)\ln(l/\delta^\star)}}$.

Fig. 1 shows the rough relationship between L and the bound given by (8).

Our bound is similar to other bounds obtained in the framework of statistical learning theory, which are loose to be used for the calculation of the exact performance of learning machine when l is not large, but can give the ideas about why a learning machine performs well and what we should do to make it behave better in some cases [36–38].

In Sections 3.2 and 3.3, we can see that the bound is able to make good explanations about and very consistent with the experimental results obtained before, and in Section 4, a prediction made by the bound is confirmed.

Please cite this article as: D. Wang, et al., An oscillation bound of the generalization performance of extreme learning machine and corresponding analysis, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.10.006

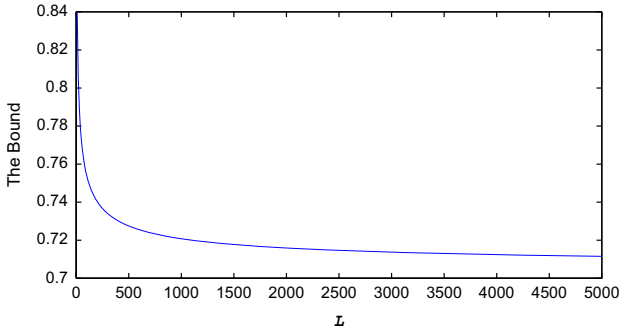

Fig. 1. The relationship between L and the bound given by (8) for $\hat{E}_{\mathbf{z}^l}(f)=0$, $C=D=1$, $\delta=\delta^\star=0.025$ and $l=5000$. It implies that overfitting can be avoided by limiting C as L increases and ELM is insensitive to L when L is large enough.

3.2. The reason why ELM is insensitive to the number of hidden nodes

Based on the analysis above, we discuss about the answer to the Question 1 in [2] that is why the generalization performance of ELM is insensitive to L in this part.

For fixed θ^L, if we use the first $\hat{L}$ parameters to build ELM with $\hat{L}$ hidden nodes where $\hat{L}\le L$, the larger the $\hat{L}$ is, the smaller the minimum of empirical risk which ELM may reach. (For example, let $L_1\le L_2$. If $\beta_{L_1+1}=\beta_{L_1+2}=\cdots=\beta_{L_2}=0$, then ELM with L_2 hidden nodes is equivalent to ELM with L_1 hidden nodes.) But if $\hat{L}$ changes slightly, reachable minimum will not change a lot because according to Lemmas 3 and 4, it is not hard to see that the complexity of $F_C(\theta^L)$ is bounded by $(C/\sqrt{l})\sqrt{\hat{\kappa}(\mathbf{x}^l)+\sqrt{(2/L)\ln(l/\delta^\star)}}$ which is insensitive to L for a fixed $\mathbf{z}^l$. The ratio of balance between the empirical risk and the norm of output weights which is related to $\sqrt{\hat{\kappa}(\mathbf{x}^l)+\sqrt{(2/L)\ln(l/\delta^\star)}}$ is also insensitive to L, that is, the C of the output function of ELM is insensitive to L. So any term on the right of inequality (5) is insensitive to L for the output function of ELM. That is why ELM is insensitive to L.

3.3. The consistence between the bound and the experimental results obtained before

The bound given by (5) reveals that for fixed $\mathbf{z}^l$, when L gets larger, $\sqrt{(2/L)\ln(l/\delta^\star)}$ in $\sqrt{\hat{\kappa}(\mathbf{x}^l)+\sqrt{(2/L)\ln(l/\delta^\star)}}$ gets smaller, which means that the influence on the performance from $\delta^\star$ gets less and the randomness from $\mathbf{\Theta}^L$ gets lost gradually. This phenomenon is shown in [25,26]. The gradual loss of randomness from $\mathbf{\Theta}^L$ also implies that when the number of hidden nodes L is large enough, tuning the parameters of hidden nodes is not necessary, which is the main idea of ELM.

The bound also shows us that as long as there is a proper limit of C or equally, a reasonable limit of the capacity of the function class, overfitting can be prevented while L gets very large or even approaches infinity, which is indirectly proven by the results in [22] where the ELM kernel with very large L was used for support vector machine, and in [23] where the ELM kernel with L approaching infinity was used for support vector regression.

In [3], the experiments showed that if we tune λ properly, which is equivalent to a proper limit of C, ELM can perform well when L equals 1000 other than overfit the data, which is a direct proof of the case that L can be very large.

All the experiments above show the validity of the bound.

According to Lemma 4, we can see that when L approaches infinity and $\kappa(\mathbf{x}_i,\mathbf{x}_j)$ exists, the ELM kernel $\kappa_{\theta^L}(\mathbf{x}_i,\mathbf{x}_j)$ approaches $\kappa(\mathbf{x}_i,\mathbf{x}_j)$ and the learning process of ELM is actually a kernel ridge regression [31,39] on the training set utilizing the kernel $\kappa(\mathbf{x}_i,\mathbf{x}_j)$.

In this paper, we refer to ELM with finite hidden nodes as original ELM and ELM with infinite hidden nodes (i.e. L approaches infinity) as ELM with infinite hidden nodes whose output function is the same as that of the ridge regression using kernel $\kappa(\mathbf{x}_i,\mathbf{x}_j)$ on the training set. To prove that overfitting can be avoided when L approaches infinity directly, we conducted the experiments in Section 4.

4. Experiments

As (1) satisfies our assumption that the expectation of the ELM kernel exists, we used $g(\mathbf{w}_j^T\mathbf{x}+b_j)$ as the activation function in our experiments where $g(t)=(2/\sqrt{\pi})\int_0^t e^{-z^2}\,dz$ for $t\in\mathbf{R}$, and each component of $\mathbf{W}$ and B obeys Gaussian distribution with mean 0 and variance σ^2. This activation function has been used to analyze the behavior of ELM and its variants for experiments in [23,25,26].

The output function of original ELM is

$$f(\mathbf{x})=[\kappa_{\theta^L}(\mathbf{x},\mathbf{x}_1),\kappa_{\theta^L}(\mathbf{x},\mathbf{x}_2),\ldots,\kappa_{\theta^L}(\mathbf{x},\mathbf{x}_l)]\left(\frac{\mathbf{I}}{\lambda}+\mathbf{\Pi}_{l\times l}\right)^{-1}\mathbf{Y},$$

where $\mathbf{\Pi}_{l\times l}=[\kappa_{\theta^L}(\mathbf{x}_i,\mathbf{x}_j)]_{l\times l}$ and the meta-parameters of it are L, λ and σ.

The output function of ELM with infinite hidden nodes is

$$f(\mathbf{x})=[\kappa(\mathbf{x},\mathbf{x}_1),\kappa(\mathbf{x},\mathbf{x}_2),\ldots,\kappa(\mathbf{x},\mathbf{x}_l)]\left(\frac{\mathbf{I}}{\lambda}+\mathbf{\Phi}_{l\times l}\right)^{-1}\mathbf{Y}$$

where $\mathbf{\Phi}_{l\times l}=[\kappa(\mathbf{x}_i,\mathbf{x}_j)]_{l\times l}$ and the meta-parameters of it are λ and σ.

The experiments in [23] showed that $\kappa(\mathbf{x}_i,\mathbf{x}_j)$ is insensitive to σ for support vector regression when σ is large enough. We conjecture that $\kappa(\mathbf{x}_i,\mathbf{x}_j)$ is also insensitive to σ for ELM. So the experiments in this section are conducted to prove that for the ELM with infinite hidden nodes, it is unnecessary to tune σ when σ is large enough, and the ELM with infinite hidden nodes can match the original ELM as long as λ is tuned properly, i.e. overfitting can be avoided when L approaches infinity.

4.1. Experimental setting

With reference to the experiments conducted in [23], we compare 8 learning machines which are 7 ELMs with infinite hidden nodes whose σ's are fixed and different from each other $(\sigma=10^{-3},\sigma=10^{-2},\sigma=10^{-1},\sigma=1,\sigma=10,\sigma=10^2,\sigma=10^3)$ and the original ELM. All the simulations were carried out in Matlab R2013b and 15 data sets from UCI machine Learning Repository [40] (Servo, Urban, CPU, Leaf, Dermatology, Auto MPG, Housing, Cancer, Noise, Handwritten Digit, Multiple Features, Chess and Abolone) and Statlib [41] (Balloon, Space-ga) were used. The first k natural numbers was used to encode the nominal attributes with k different values and the details about the data sets are given in Table 1.

We first normalized all input variables into $[-1,1]$. For each data set, ten experiments were conducted. For each experiment, the data set was randomly split into two parts. One-third was used as the test set for the 8 machines built on the remaining set in which 5-fold cross-validation was performed to select the meta-parameters where λ was chosen from the range $\{2^{-5},2^{-4},\ldots,2^{16}\}$ (22 values) and L was chosen from the range $\{10,20,30,50,70,100,120,150,200,300,500,700,900,1000,1100,1200,1500,2000,3000,5000\}$ (20 values) almost like the ranges used in [5]. The mean squared error was the criterion used to compare models and the mean of ten test results and corresponding standard deviation which are denoted by MSE and DEV respectively are given in Table 2.

Please cite this article as: D. Wang, et al., An oscillation bound of the generalization performance of extreme learning machine and corresponding analysis, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.10.006

Table 1
Details about the data sets.

Short name	Size	Dimensionality	Full name
Servo	167	4	Servo data set
Urban	168	147	Urban land cover data set
CPU	209	6	Computer hardware data set
Leaf	340	14	Leaf data set
Dermatology	358	33	Dermatology data set
Auto MPG	392	7	Auto MPG data set
Housing	506	13	Housing data set
Cancer	569	30	Breast Cancer Wisconsin (Diagnostic) data set
Noise	1503	5	Airfoil self-noise data set
Handwritten Digit	1593	256	Semeion handwritten digit data set
Multiple Features	2000	649	Multiple features data set
Balloon	2001	1	Balloon
Space-ga	3107	6	Space-ga
Chess	3196	36	Chess (King-Rook vs. King-Pawn) data set
Abolone	4177	7	Abolone data set

Table 2
Experimental results.

Data set	Original ELM	ELM with Infinite Hidden Nodes						
		$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 10^{-1}$	$\sigma = 1$	$\sigma = 10$	$\sigma = 10^{2}$	$\sigma = 10^{3}$
Servo								
MSE	0.0392	0.1305	0.1144	0.0495	**0.0334**	**0.0326**	**0.0335**	**0.0337**
DEV	0.0043	0.0066	0.0035	0.0042	0.0027	0.0031	0.0030	0.0030
Urban								
MSE	0.1335	0.2062	0.2055	0.1500	**0.1265**	**0.1259**	**0.1257**	**0.1257**
DEV	0.0047	0.0052	0.0051	0.0054	0.0048	0.0047	0.0046	0.0046
CPU								
MSE	0.0166	0.0222	0.0206	**0.0134**	**0.0140**	**0.0140**	**0.0140**	**0.0140**
DEV	0.0036	0.0029	0.0030	0.0029	0.0032	0.0029	0.0029	0.0029
Leaf								
MSE	0.1710	0.3663	0.3309	0.2098	**0.1584**	**0.1599**	**0.1596**	**0.1599**
DEV	0.0124	0.0054	0.0104	0.0078	0.0077	0.0098	0.0091	0.0090
Dermatology								
MSE	0.0352	0.0611	0.0605	0.0370	**0.0313**	**0.0317**	**0.0319**	**0.0320**
DEV	0.0023	0.0025	0.0026	0.0024	0.0023	0.0021	0.0021	0.0021
Auto MPG								
MSE	0.0194	0.0333	0.0317	0.0202	**0.0191**	**0.0190**	**0.0191**	**0.0191**
DEV	0.0011	0.0011	0.0011	0.0010	0.0011	0.0011	0.0011	0.0010
Housing								
MSE	0.0215	0.0544	0.0461	**0.0202**	**0.0182**	**0.0193**	**0.0198**	**0.0199**
DEV	0.0019	0.0028	0.0024	0.0014	0.0014	0.0014	0.0014	0.0014
Cancer								
MSE	0.1327	0.2664	0.2431	0.1750	**0.1266**	**0.1271**	**0.1273**	**0.1272**
DEV	0.0062	0.0067	0.0059	0.0076	0.0067	0.0068	0.0067	0.0067
Noise								
MSE	0.0256	0.0713	0.0686	0.0433	**0.0283**	**0.0206**	**0.0161**	**0.0171**
DEV	0.0017	0.0013	0.0013	0.0007	0.0009	0.0014	0.0019	0.0020
Handwritten digit								
MSE	0.1590	0.1920	0.1664	**0.1241**	**0.1361**	**0.1384**	**0.1386**	**0.1386**
DEV	0.0035	0.0038	0.0055	0.0029	0.0031	0.0032	0.0032	0.0032
Multiple features								
MSE	0.0330	0.0404	**0.0319**	**0.0233**	**0.0287**	**0.0298**	**0.0299**	**0.0300**
DEV	0.0011	0.0010	0.0008	0.0009	0.0011	0.0011	0.0011	0.0011
Balloon								
MSE	0.0071	0.0111	0.0106	0.0077	**0.0073**	**0.0071**	**0.0071**	**0.0071**
DEV	0.0001	0.0003	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001
Space-ga								
MSE	0.0042	0.0093	0.0068	0.0051	**0.0041**	**0.0042**	**0.0043**	**0.0043**
DEV	0.0001	0.0004	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001
Chess								
MSE	0.0787	0.3764	0.3657	**0.0671**	**0.0641**	**0.0769**	**0.0784**	**0.0786**
DEV	0.0018	0.0025	0.0026	0.0014	0.0016	0.0019	0.0019	0.0019
Abolone								
MSE	0.0227	0.0263	0.0252	0.0231	**0.0227**	**0.0226**	**0.0225**	**0.0225**
DEV	0.0003	0.0004	0.0003	0.0004	0.0003	0.0003	0.0003	0.0003

Please cite this article as: D. Wang, et al., An oscillation bound of the generalization performance of extreme learning machine and corresponding analysis, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.10.006

4.2. Experimental results and analysis

The results of the ELMs with infinite hidden nodes which are better than or similar to those of the original ELM are indicated in bold.

Table 2 shows that if $\sigma \geq 1$, the results of the ELMs with infinite hidden nodes are as good as or even better than the original ELM and not significantly different from each other. Like the conclusion drawn by [23], it can be concluded that the performance of the ELM with infinite hidden nodes is insensitive to parameter σ and σ can be assigned 1 or 10 without tuning it.

The results also show that the learning ability of the ELM with infinite hidden nodes can match that of the original ELM and effectively avoid overfitting, which confirms the prediction. This gives another experimental proof of the validity of the bound.

5. Conclusions

An oscillation bound of the generalization performance of ELM is obtained in the framework of statistical learning theory and under the assumption that the expectation of the ELM kernel exists. The bound is able to explain the reason why ELM is insensitive to the number of hidden nodes well and consistent with the previous experimental results. It predicts that overfitting can be avoided when L approaches infinity. The prediction has been confirmed by our experiments in which one kind of activation function with proper distribution of parameters was used. The experimental results not only show the validity of the bound, but also prove that the learning ability of the ELM with infinite hidden nodes can match that of the original ELM.

The results also suggest that the ELM with infinite hidden nodes in the experiments is insensitive to the kernel parameter when the parameter is large enough and can get similar performance compared to the original ELM, which means that only one meta-parameter of the ELM with infinite hidden nodes need to be tuned.

The inspiration of our work is the insight that ELM is a kind of kernel method and the reason why it performs well can be found in the Law of Large Numbers.

Many people may be confused that how a learning machine can perform well when there is randomness in itself and what if the parameters are all the same. When the structure of the machine is simple, e.g. $L=5$, everyone should worry about that. But if L equals a thousand and even more, the event that the parameters are all the same just does not happen! People in various application fields hate randomness which makes them cannot get the exact answers as randomness is from nature and out of control, e.g. they cannot collect enough samples to do analysis subject to nature conditions, they do not know whether the samples are independently and identically distributed and so on. But if the randomness is under control, for example, it is introduced to ELM by ourselves, the case is not the same. Because we can sample as many as we need, know the nature of the randomness and more importantly, we can always get help from the Law of Large Numbers, which again takes us back to the determinacy! Is it a waste of time? The answer is no! All the theory and application study of ELM besides our work have shown that with this method, ELM not only even saves time for us but also shows its powerful learning ability. ELM reminds us that what powerful tools Random methods and the Law of Large Numbers are!

Acknowledgment

The authors would like to thank the anonymous editor and reviewers for their valuable comments and suggestions which improved this work.

The authors are also thankful to Application Basis and Cutting-edge Technology Research Projects of Tianjin City, China (14JCYBJC21800), The 2014 Annual China Public Industry (Meteorological) Research Project (GYHY201406004) and China Meteorological Administration: Development and Application of the Software System for a New Generation Weather Radar Building Business for support.

References

[1] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: 2004 IEEE International Joint Conference on Neural Networks, 2004. Proceedings, Budapest, Hungary, vol. 2, IEEE, 2004, pp. 985–990.
[2] G.-B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybern. 2 (2) (2011) 107–122.
[3] 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.
[4] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.
[5] J. Chorowski, J. Wang, J.M. Zurada, Review and performance comparison of svm-and elm-based classifiers, Neurocomputing 128 (2014) 507–516.
[6] J.-h. Zhai, H.-y. Xu, X.-z. Wang, Dynamic ensemble extreme learning machine based on sample entropy, Soft Comput. 16 (9) (2012) 1493–1502.
[7] D. Wang, M. Alhamdoosh, Evolutionary extreme learning machine ensembles with size control, Neurocomputing 102 (2013) 98–110.
[8] R. Savitha, S. Suresh, N. Sundararajan, Fast learning circular complex-valued extreme learning machine (cc-elm) for real-valued classification problems, Inf. Sci. 187 (2012) 277–290.
[9] X. Xue, M. Yao, Z. Wu, J. Yang, Genetic ensemble of extreme learning machine, Neurocomputing 129 (2014) 175–184.
[10] H.-G. Han, L.-D. Wang, J.-F. Qiao, Hierarchical extreme learning machine for feedforward neural network, Neurocomputing 128 (2014) 128–135.
[11] J. Zhao, Z. Wang, D.S. Park, Online sequential extreme learning machine with forgetting mechanism, Neurocomputing 87 (2012) 79–89.
[12] J. Cao, Z. Lin, G.-B. Huang, Self-adaptive evolutionary extreme learning machine, Neural Process. Lett. 36 (3) (2012) 285–305.
[13] J. Cao, Z. Lin, G.-B. Huang, N. Liu, Voting based extreme learning machine, Inf. Sci. 185 (1) (2012) 66–77.
[14] Y. Wang, F. Cao, Y. Yuan, A study on effectiveness of extreme learning machine, Neurocomputing 74 (16) (2011) 2483–2490.
[15] W. Xi-Zhao, S. Qing-Yan, M. Qing, Z. Jun-Hai, Architecture selection for networks trained with extreme learning machine using localized generalization error model, Neurocomputing 102 (2013) 3–9.
[16] M. Xia, Y. Zhang, L. Weng, X. Ye, Fashion retailing forecasting based on extreme learning machine with adaptive metrics of inputs, Knowl. Based Syst. 36 (2012) 253–259.
[17] R. Minhas, A. Baradarani, S. Seifzadeh, Q. Jonathan Wu, Human action recognition using extreme learning machine based on visual vocabularies, Neurocomputing 73 (10) (2010) 1906–1917.
[18] G.-B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (1) (2010) 155–163.
[19] F. Chen, T. Ou, Sales forecasting system based on gray extreme learning machine with taguchi method in retail industry, Expert Syst. Appl. 38 (3) (2011) 1336–1345.
[20] W. Zong, G.-B. Huang, Y. Chen, Weighted extreme learning machine for imbalance learning, Neurocomputing 101 (2013) 229–242.
[21] Q. Liu, Q. He, Z. Shi, Extreme support vector machine classifier, in: Advances in Knowledge Discovery and Data Mining, Springer, Berlin, 2008, pp. 222–233.
[22] B. Frénay, M. Verleysen, et al., Using svms with randomised feature spaces: an extreme learning approach, in: ESANN, 2010.
[23] B. Frénay, M. Verleysen, Parameter-insensitive kernel in extreme learning for non-linear support vector regression, Neurocomputing 74 (16) (2011) 2526–2531.
[24] G.-B. Huang, An insight into extreme learning machines: random neurons, random features and kernels, Cogn. Comput. (2014) 1–15.
[25] E. Parviainen, J. Riihimäki, Y. Miche, A. Lendasse, Interpreting extreme learning machine as an approximation to an infinite neural network, in: KDIR, Citeseer, 2010, pp. 65–73.
[26] E. Parviainen, J. Riihimäki, A connection between extreme learning machine and neural network kernel, in: Knowledge Discovery, Knowledge Engineering and Knowledge Management, Springer, Berlin, Heidelberg, 2013, pp. 122–135.
[27] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.
[28] M. Leshno, V.Y. Lin, A. Pinkus, S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Netw. 6 (6) (1993) 861–867.
[29] J. Park, I.W. Sandberg, Universal approximation using radial-basis-function networks, Neural Comput. 3 (2) (1991) 246–257.
[30] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (16) (2007) 3056–3062.

Please cite this article as: D. Wang, et al., An oscillation bound of the generalization performance of extreme learning machine and corresponding analysis, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.10.006

[31] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, New York, 2004.
[32] V.N. Vapnik, V. Vapnik, Statistical Learning Theory, vol. 2, Wiley, New York, 1998.
[33] M. Anthony, P.L. Bartlett, Neural Network Learning: Theoretical Foundations, Cambridge University Press, New York, 2009.
[34] P.L. Bartlett, S. Mendelson, Rademacher and Gaussian complexities: risk bounds and structural results, J. Mach. Learn. Res. 3 (2003) 463–482.
[35] M. Hermans, B. Schrauwen, Recurrent kernel machines: computing with infinite echo state networks, Neural Comput. 24 (1) (2012) 104–133.
[36] C.J. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov. 2 (2) (1998) 121–167.
[37] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Stat. (1998) 1651–1686.
[38] L. Wang, M. Sugiyama, Z. Jing, C. Yang, Z.-H. Zhou, J. Feng, A refined margin analysis for boosting algorithms via equilibrium margin, J. Mach. Learn. Res. 12 (2011) 1835–1863.
[39] C. Saunders, A. Gammerman, V. Vovk, Ridge regression learning algorithm in dual variables, in: (ICML-1998) Proceedings of the 15th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, 1998, pp. 515–521.
[40] K. Bache, M. Lichman, UCI machine learning repository, 2013. URL ⟨http://archive.ics.uci.edu/ml⟩.
[41] P. Vlachos, Statlib datasets archive. URL ⟨http://lib.stat.cmu.edu/datasets⟩.

Di Wang received his B.E. degree in electrical engineering and its automation from Tianjin University, China, in 2012. He is now a graduate student at Department of Electrical Engineering and Automation, Tianjin University. His current research interests include extreme learning machine, kernel method and machine learning.

Ping Wang is a professor at Department of Electrical Engineering and Automation, Tianjin University, China. She is a PhD supervisor in control science and engineering. Her research interests include pattern recognition and its application, image understanding and moving objects tracking.

Yan Ji received her B.E degree in information management and information system from Nankai University, China, in 2012. She received her M.S. degree in management science and engineering from Nankai University in 2014. Her research interests include machine learning and data mining.

Please cite this article as: D. Wang, et al., An oscillation bound of the generalization performance of extreme learning machine and corresponding analysis, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.10.006