Although the whole article is meaningless, we primarily comment on the abstract and at a few other places in the article. This article also makes use of words such as 'may' and 'analysis' to make favorable statements about ELM and unfavorable statements about all other competing methods in an unscientific manner without extensive experimental results nor accurate proofs.

What are Extreme Learning Machines? Filling the Gap between Frank Rosenblatt's Dream and John von Neumann's Puzzle with Stolen Goods

There was no puzzle in 2004 when ELM name was created. Lowe, Broomhead, Schmidt and Pao presented randomized input weights with closed-form solution for output weights in 1988, 1992, and 1994 respectively.

Huang created a new name by excluding the closely related papers in his ELM-RBF & ELM-SLFN papers. Huang's papers had tiny variation from RVFL (no direct link) and RBF (randomizing impact factors). These Huang's novelties make the original methods perform worse.

Guang-Bin Huang

All these theories have been presented by other researchers for MLP, RVFL, RBF, etc. many-many years before ELM was created in 2004.

***Abstract*—The emergent machine learning technique - Extreme Learning Machines (ELMs) - has become a hot area of research over the past years, which is attributed to the growing research activities and significant contributions made by numerous researchers around the world. Recently, it has come to our attention that a number of misplaced notions and misunderstandings are being dissipated on the relationships between ELM and some earlier works. This paper wishes to clarify that: i) ELM theories manage to address the open problem which has puzzled the neural networks, machine learning and neuroscience communities for 60 years: whether hidden nodes / neurons need to be tuned in learning, and proved that in contrast to the common knowledge and conventional neural network learning tenets, hidden nodes / neurons do not need to be iteratively tuned in wide types of neural networks and learning models (Fourier series, biological learning, etc). Unlike ELM theories, none of those earlier works provides theoretical foundations on feedforward neural networks with random hidden nodes; ii) ELM is proposed for both generalized single hidden layer feedfoward network and multi hidden layers feedforward networks; iii) Homogeneous architecture based ELM is proposed for feature learning, clustering, regression and (binary / multi-class) classification. iv) Compared to ELM, SVM and LS-SVM tend to provide suboptimal solutions, and SVM and LS-SVM do not consider feature representations in hidden layers of multi layers of networks either.**

***Index Terms*—Extreme learning machine, random vector functional link, QuickNet, radial basis function network, feedforward neural network, randomness.**

RVFL and Schmidt's Randomized NN all can be easily extended without unethically creating the ELM name.

SVM clearly outperforms ELM as presented in the recent JMLR article.

I. Introduction

DESPITE that the relationships and differences between Extreme Learning Machines (ELM) and those earlier works (e.g., Schmidt, *et al.* [1] and RVFL [2]) have been clarified in [3]–[8], recently several researchers insist that those earlier works are the "origins" of ELM and ELM essentially the same as those earlier works, and thus, further claimed that it is not necessary to have a new term Extreme Learning Machines (ELM). This paper wishes to clarify the essential elements of ELMs which may have been overlooked in the past years.

We prefer to avoid the word "origin" in this paper as 1) it may be really difficult to show which is the "true" "origin" in a research area as most works are related to each other; 2) it may cause unnecessary inconvenience or potential "controversy" among those listed as "origins" and other pioneering works which may have been missing in discussions. *The ultimate goal of research is to find the truth of natural phenomena and to move research forward instead of arguing for being listed as "origins."* Otherwise, many earlier works should not have had their own terms, and instead almost all should simply have been called "feedforward neural networks" or should even simply go back to Frank Rosenblatt's "perceptrons" [9]. Such misunderstandings on ignoring the needs of having new terms would actually discourage researchers' creativeness and their spirit of telling the truth and differences in research. Similarly, there is nothing wrong to have new terms for the variants of ELM (with Fourier series nodes) with significant extensions (such as Random Kitchen Sink (RKS) [10], RKS' further extension - FastFood [11] and Convex Network [12]) as well as ELM with LMS referred to as No-Prop algorithm [13].

Ge[illegible] *'Extreme' here means to move beyond conventional artificial learning techniques and to move toward brain alike learning. ELM aims to break the barriers between the conventional artificial learning techniques and biological learning mechanism. 'Extreme learning machine (ELM)' represents a suite of machine learning techniques in which hidden neurons need not be tuned with the consideration of neural network generalization theory, control theory, matrix theory, and linear system theory."*

In order to have clearer understanding on ELM, it is better to analyze ELM in the aspects of its philosophy, theories, network architecture, network neuron types, and its learning objectives and algorithms.

II. ELM's Beliefs, Philosophy and Objectives

ELM works start from our intuitive belief on biological learning and neural networks generalization performance theories [14]. Further development of ELM works are also built on top of Frank Rosenblatt's multi-layer 'perceptrons" [9], SVM [15], LS-SVM [16], Fourier series, linear systems, numeral methods, matrix theories, etc but with essential extensions.

Frank Rosenblatt [9] believes that multi layer feedforward networks (perceptrons) can enable computers to "walk, talk, see, write, reproduce itself and be conscious of its existence."[1]. Minsky and Papert [17] do not believe that perceptrons have such learning capabilities by giving a counter example showing that a perceptron without having hidden layers even could

G.-B. Huang is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798 (E-Mail: egbhuang@ntu.edu.sg).

[1]http://en.wikipedia.org/wiki/Perceptron

Researchers who plagiarise others work are possibly allergic to the word 'origin'. But, the Nobel committee every year identifies the origin of important works to award the TRUE inventors. Further, identifying the origins of randomized neural networks is not as hard as discovering the origin of the universe.

not handle the simple XOR problem. Such a counter example made many researchers run away from artificial neural networks, and finally resulted in the "Artificial Intelligence (AI) Winter" in 1970s. To our understanding, there is an interesting issue between Rosenblatt's dream and Minsky's counter example. Rosenblatt may not be able to give efficient learning algorithms in the very beginning of neural networks research. Rosenblatt's perceptron is a multi layer feedforward network. In many cases, a feedforward network with input and output layers but without hidden layers is considered as a two-layer perceptron, which were actually used in Minsky and Papert [17]. However, a feedforward network with input and output layers but without hidden layers seems like a "brain" which has input layers (eyes, noses, etc) and output layers (motor sensors, etc) but without "central neurons." Obviously such a "brain" is an empty shell and has no "learning and cognition" capabilities at all. However, Rosenblatt and Minsky's controversy also tells the truth that one small step of development in artificial intelligence and machine learning may request one or several generations' great efforts. Their professional controversy[2] may turn out to indirectly inspire the reviving of artificial neural network research in the end.

Thus, there is no doubt that neural networks research revives after hidden layers are emphasized in learning since 1980s. However, an immediate dilemma in neural network research is that: since hidden layers are important and necessary conditions of learning, by default expectation and understanding of neural network research community, hidden neurons of all networks need to be tuned. Thus, since 1980s tens of thousands of researchers from almost every conner of the world have been working hard on looking for learning algorithms to train various types of neural networks mainly by tuning hidden layers. Such a kind of "confusing" research situation turned out to force us to seriously ask several questions as early as 1995 [18]:

1) Do we really need to spend so much manpower and great effort on finding learning algorithms and manually tuning parameters for different neural networks and applications? However, obviously, in contrast there is no "pygmy" sitting in biological brains and tuning parameters there.
2) Do we really need to have different learning algorithms for different types of neural networks in order to achieve good learning capabilities of feature learning, clustering, regression and classification?
3) Why are biological brains more "efficient" and "intelligent" than those machines / computers embedded with artificially designed learning algorithms?
4) Are we able to address John von Neumann's puzzle[3] [19], [20] why "an imperfect neural network, containing many random connections, can be made to perform reliably those functions which might be represented by idealized wiring diagrams?"

[2]Professional controversies should be advocated in academic and research environments, however, irresponsible anonymous attack which intends to destroy harmony research environment and does not help maintain healthy controversies should be refused.

[3]John von Neumann was also acknowledged as a "Father of Computers."

No solutions to the above mentioned problems were found until 2003 after many years of efforts spent. Finally it was found that the key "knot" in the abovementioned open problems is that

1) The counter example given by Minsky and Papert [17] shows that hidden layers are necessary.
2) Earlier neural networks theories on universal approximation capabilities (e.g, [21], [22]) are also built on the assumption that hidden neurons need to be tuned during learning.
3) Thus, naturally and reasonably speaking, hidden neurons need to be tuned in artificial neural networks.

In order to address the abovementioned open problems, one has to untie the key "knot," that is, for wide types of networks (artifical neural networks or biological neural networks whose network architectures and neuron modeling are even unknown to human being), hidden neurons are important but do not need to be tuned.

Such beliefs and philosophy in both machine learning and biological learning finally result in the new techniques referred to Extreme Learning Machines (ELM) and related ELM theories. As emphasized in Huang, *et al.* [6], 'Extreme' means to move beyond conventional artificial learning techniques and to move toward brain alike learning. ELM aims to break the barriers between the conventional artificial learning techniques and biological learning mechanism. 'Extreme learning machine (ELM)' represents a suite of machine learning techniques (including *single hidden layer feedforward networks* and *multi hidden-layers feedforward networks*) in which hidden neurons do not need to be tuned with the consideration of neural network generalization theory, control theory, matrix theory, and linear system theory. To randomly generate hidden nodes is one of the typical implementations which ensures that "hidden neurons do not need to be tuned" in ELM, however, there also exist many other implementations such as kernels [6], [23], SVD, and local receptive fields [8], etc. We believe that ELM reflects the truth of some biological learning mechanisms. Its machine-based learning efficiency was confirmed in 2004 [24], its universal approximation capability (for "generalized SLFNs" in which a hidden node may be a subnetwork of several nodes and / or with almost any nonlinear piecewise continuous neurons (although their exact mathematical modeling /formula / shapes may be unknown to human beings)) was rigorously proved in theory in 2006-2008 [5], [25], [26]. Its concrete biological evidence subsequently appears in 2011-2013 [27]–[30].

ELM targets at not only "generalized" single hidden layer feedforward networks but also "generalized" multi hidden layers feedforward networks in which a node may be a subnetwork consisting of other hidden nodes [5], [8], [26]. Single hidden layer of ELM also covers wide types of neural networks including but not limited to sigmoid networks and RBF networks. (Refer to Subsection IV-A for details.)

Compression, feature learning, clustering, regression and classification are fundamental to machine learning and machine intelligence. ELM aims to implement these five fundamental operations / roles of learning in homogeneous ELM architectures (cf. Figure 1).

Please do not just listen and refuse. Please investigate yourself to find out the truth. We have offered a lot of evidences, annotated PDF files, etc. Indeed, the origin of ELM is unhealthy and intends to destroy harmonious research environment.

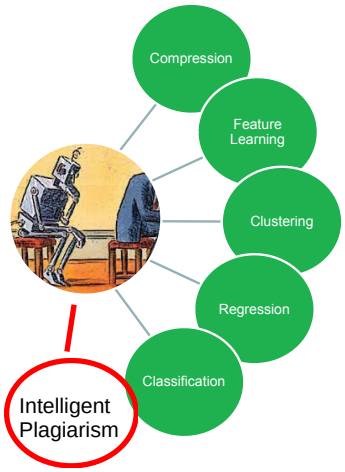
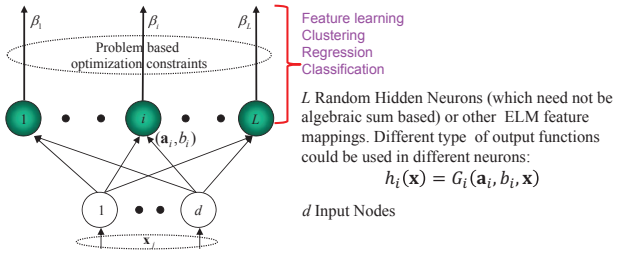
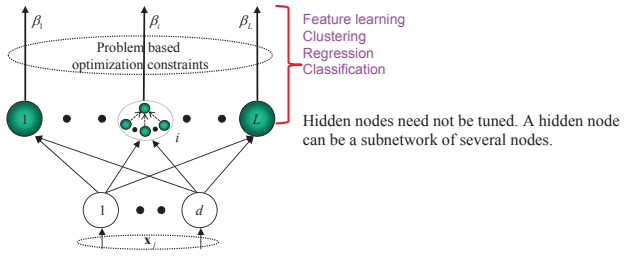


Fig. 1. Fundamental operations / roles of ELM. Courtesy to the anonymous designer who provides the robot icon in Internet



(a) ELM slice / feature mapping with fully connected random hidden nodes



(b) ELM slice / feature mapping with subnetworks

Fig. 2. ELM theories [5], [25], [26] show wide types of hidden nodes can be used in each ELM slice (ELM feature mapping) in which a hidden node can be a subnetwork of several nodes.

This is a lie: (1). ELM "random selection also depends on data, since you normalized both your weights and data to 1. (2). If you select weights, e.g., from 1 to 1 million, where the data is only from 1 to 10, you may make it work, but in a stupid way.

Although there are few attempts on random sigmoid hidden neurons and / or RBF neurons in 1950s-1990s [1], [2], [9], [31]–[33], this kind of implementations did not really "take off" except for RVFL [34] due to several reasons:

1) The common understanding and tenet is that hidden neurons of various types of neural networks need to be tuned.
2) There lacks of theoretical analysis except for RVFL.
3) There lacks of strong motivation from biological learning except for Rosenblatt's perceptron.

ELM theories managed to address the challenging issue: "*whether wide types of neural networks (including biological neural networks) with wide types of hidden nodes / neurons (almost any nonlinear piecewise continuous nodes) can be randomly generated.*" Although ELM aims to deal with both single hidden layer feedforward networks (SLFNs) and multi hidden layer feedforward networks, its theories have mainly focused on SLFN cases in the past 10 years.

A. Universal approximation capability

Strictly speaking, none of those earlier works (e.g., Baum [31] and Schmidt, *et al.* [1], RVFL [2], [32]) has addressed in theory whether random hidden nodes can be used in their specific sigmoid or RBF networks, let alone the wide type of networks covered by ELM theories. Lowe [35]'s RBF network does not use the random impact factor although the centers of their RBF nodes are randomly generated. One has to adjust impact factors based on applications. In other words, semi-random RBF nodes are used in RBF network [35]. Detail analysis has been given in Huang [3].

Both Baum [31] and Schmidt, *et al.* [1] focus on empirical simulations on specific network architectures (a specific case of ELM models).[4] To the best of our knowledges, both earlier works do not have theoretical analysis, let alone the rigorous theoretical proof. Although intuitively speaking, Igelnik and Pao [32] tries to prove the universal approximation capability of RVFL, as analyzed in [4], [8], actually Igelnik and Pao [32] only proves RVFL's universal approximation capability when *semi-random sigmoid and RBF hidden nodes* are used, that is, the input weights $\mathbf{a}_i$ are randomly generated while the hidden node biases b_i are calculated based on the training samples $\mathbf{x}_i$ and the input weights $\mathbf{a}_i$. (Refer to Huang, *et al.* [4] for detail analysis.)

In contrast, ELM theories have shown that almost any nonlinear piecewise continuous random hidden nodes (including sigmoid and RBF nodes mentioned in those earlier works, but also including wavelet, Fourier series, biological neurons, etc) can be used in ELM and the resultant networks have universal approximation capabilities [5], [25], [26]. Unlike the semi-random sigmoid and RBF hidden nodes used in the proof of RVFL [32] in which some parameters are not randomly generated, the physical meaning of *random hidden nodes* in ELM theories is that all the parameters of the hidden nodes are randomly generated independently from the training samples, e.g., both random input weights $\mathbf{a}_i$ and biases b_i for additive hidden nodes, or both centers $\mathbf{a}_i$ and impact factor b_i for RBF networks, parameters for Fourier series and wavelets, etc. It is ELM theories first time to show that all the hidden nodes / neurons can be not only independent from training samples but also independent from each other in wide types of neural networks and mathematical series / expansions as well as in biological learning mechanism [5], [6], [25], [26].

Definition 3.1: [5], [25], [26] A hidden layer output mapping $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \cdots, h_L(\mathbf{x})]$ is said to be an ELM random feature mapping if all its hidden node parameters are randomly generated according to any continuous sampling distribution probability, where $h_i(\mathbf{x}) = G_i(\mathbf{a}_i, b_i, \mathbf{x})$, $i = 1, \cdots, L$ (the number of neurons in the hidden layer)..

[4]Schmidt, *et al.* [1] only reported some experimental results on three synthetic toy data as usually done by many researchers in 1980s-1990s, however, it may be difficult for machine learning community to make concrete conclusions based on experimental results on toy data in most cases.

1. Even if you proved correctly (or incorrectly) the universal approximation of RVFL, you don't need to create another name ELM to steal the ownership of other's work. 2. Experimental comparisons show that RVFL (with input-output direct links) is superior than without direct links ("ELM" or Schmidt NN)

a

Different hidden nodes may have different output functions G_i. In most applications, for the sake of simplicity, same output functions can be chosen for all hidden nodes, that's, $G_i = G_j$ for all $i, j = 1, \cdots, L$.

Theorem 3.1: Universal approximation capability [5], [25], [26]: Given any nonconstant piecewise continuous function as the activation function, if tuning the parameters of hidden neurons could make SLFNs approximate any target function $f(\mathbf{x})$, then the sequence $\{h_i(\mathbf{x})\}_{i=1}^{L}$ can be randomly generated according to any continuous distribution probability, and $\lim_{L\to\infty} \| \sum_{i=1}^{L} \boldsymbol{\beta}_i h_i(\mathbf{x}) - f(\mathbf{x}) \| = 0$ holds with probability one with appropriate output weights $\boldsymbol{\beta}$.

B. Classification capability

In addition, ELM theories also prove the classification capability of wide types of networks with random hidden neurons, and such theories have not been studied by those earlier works.

Theorem 3.2: Classification capability [23]: Given any nonconstant piecewise continuous function as the activation function, if tuning the parameters of hidden neurons could make SLFNs approximate any target function $f(\mathbf{x})$, then SLFNs with random hidden layer mapping $\mathbf{h}(\mathbf{x})$ can separate arbitrary disjoint regions of any shapes.

IV. Single Hidden Layer Feedforward Networks vs Multi Hidden Layers Feedforward Networks

It is difficult to deal with multi hidden layers of ELM directly without having complete solutions of single hidden layer of ELM. Thus, in the past 10 years, most of ELM works have been focusing on "generalized" single hidden layer feedforward networks (SLFNs).

A. "Generalized" single hidden layer feedforward networks (SLFNs)

Schmidt, *et al.* [1] focuses on sigmoid networks and Pao, *et al.* [32] focuses on RVFL (with sigmoid or RBF nodes). Both have strict standard single hidden layers, which are not "generalized" single hidden layer feedforward networks (SLFNs) studied in ELM. Similar to SVM [15], the feedforward neural network with random weights proposed in Schmidt, *et al.* [1] requires a bias in the output node in order to absorb the system error as its universal approximation capability with random sigmoid nodes was not proved when proposed:

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i g_{\text{sig}}(\mathbf{a}_i \cdot \mathbf{x} + b_i) + b \tag{1}$$

where $g_{\text{sig}}(x) = \frac{1}{1+\exp(-x)}$. Both QuickNet and RVFL have the direct link between the input node and the output node:

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i g_{\text{sig or RBF}}(\mathbf{a}_i \cdot \mathbf{x} + b_i) + \boldsymbol{\alpha} \cdot \mathbf{x} \tag{2}$$

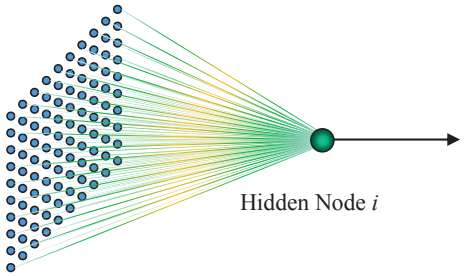
ELM is proposed for "generalized" single layer feedforward networks and mathematical series / expansions (which may not be a conventional neural network even, such as wavelet, Fourier series, etc):

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}) \tag{3}$$

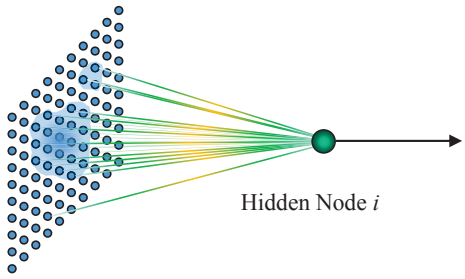
The basic ELM is for generalized SLFN, unlike the fully connected networks in those earlier works, there are three levels of randomness in ELM (Figure 3 for details):

1) Fully connected, hidden node parameters are randomly generated.
2) Connection can be randomly generated, not all input nodes are connected to a particular hidden node. Possibly only some input nodes in some local field are connected to one hidden node.
3) A hidden node itself can be a subnetwork formed by several nodes which naturally forms the local receptive fields and pooling functions, and thus results in learning local features. In this sense, some local parts of a single ELM can contain multi hidden layers.

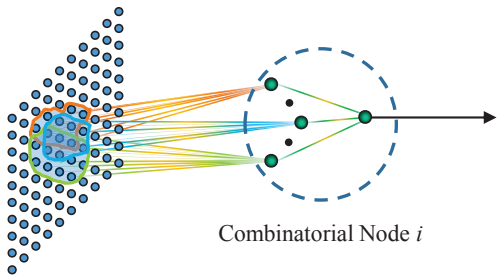
Note: Unlike Schmidt, *et al.* [1] and Pao, *et al.* [32] in which each node is a sigmoid or RBF node only, each hidden node



(a) Hidden node in full connection in ELM



(b) Hidden node in local connection / random connection in ELM



(c) Combinatorial node of several nodes in ELM

G.-B. Huang has a habit of " present" other researchers' earlier work as if these were first invented by G.-B. Huang. This figure is another example. Local connections in Fig.3(b) were first propsosed by Fukushima (Neocognitron) and LeCun (convolutional neural networks) in the 80's. Combinatorial nodes in Fig.3(c) were first proposed by Hinton (as "capsules") in ICANN 2011. These original references are not cited in this paper. According to the established pattern, these orginal references will be cited later, but still claiming novelty.

Fig. 3. ELM theories [5], [25], [26] show wide types of hidden nodes can be used and resultant networks need not be a single hidden layer feedforward networks. "Generalized SLFN" referred by ELM means a subnetwork of several nodes.

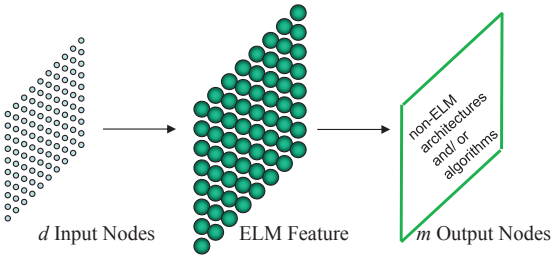


Fig. 5. ELM slice works as the input of other learning models.

in ELM can be a subnetwork of other nodes in which feature learning can be implemented efficiently. (Refer to Huang, *et al.* [8], Figure 2 and Figure 3 for details.)

According to ELM theories [5], [25], [26], ELM SLFNs include but are not limited to:

You are not the first to use these neuron types. There is nothing stopping Pao or Schmidt from using these different neurons.

1) Sigmoid networks
2) RBF networks
3) Threshold networks [36]
4) Trigonometric networks
5) Fuzzy inference systems
6) Fully complex neural networks [37]
7) High-order networks
8) Ridge polynomial networks
9) Wavelet networks
10) Fourier series [5], [6], [25], [26]
11) Bological neurons whose modeling/shapes may be unknown, etc.

B. Multi hidden layers feedforward networks

However, Unlike Schmidt, *et al.* [1] and RVFL [32] which only works for single hidden layer feedforward networks, the ultimate tenet of ELM is: hidden nodes of wide types of multi hidden layers networks do not need to be tuned (e.g., [8], [38], [39], [43])(cf. Figure 4). Although multi layers of ELM concepts have been given in ELM theories in 2007 [26], it has not been used until recently (e.g., [8], [38], [39], [43]). In essence:

1) Rosenblatt tried to transfer learned behavior from trained rats to naive rats by the injection of brain extracts,[5] which may not consider the fact that different layers of neurons may play different roles. Unlike Rosenblatt's perceptron concept, we think that it is impossible to have all the layers randomly generated. If all layers in a multi layer network are randomly generated, the useful information may not pass through two or more purely random hidden layers. However, each basic ELM can be used in each hidden layer, and hidden neurons do not need to be tuned layer wise and different layer may have different targets (in terms of ELM's five fundamental operations: compression, feature learning, clustering, regression and classification).
2) The meanings that hidden nodes do not need to be tuned are two folds:
   a) Hidden nodes may be randomly generated.
   b) Although hidden nodes do not need to be randomly generated, they need not be tuned either. For example, a hidden node in the next layer can simply be a linear sum or non-linear transform of some randomly generated nodes in the earlier layer. In this case, some nodes are randomly generated and some are not, but none of them are tuned. [8]
3) Each single ELM can deal with compression, feature learning, clustering, regression or classification. Thus, a homogeneous hierarchical blocks of ELM can be built. For example, one ELM as feature learning, the next ELM works as a classifier. In this case, we have two hidden layers of ELM, overall speaking it is not randomly generated and it is ordered, but hidden nodes in each layer do not need to be tuned (e.g, randomly generated or explicitly given/calculated).
4) ELM slices which play feature learning or clustering roles can also be used to link different learning models. Or as an entire networks, some layers are trained by ELM, and some are trained by other models (cf. Figure 6).

Schmidt can also do these things, so can RVFL, only better!

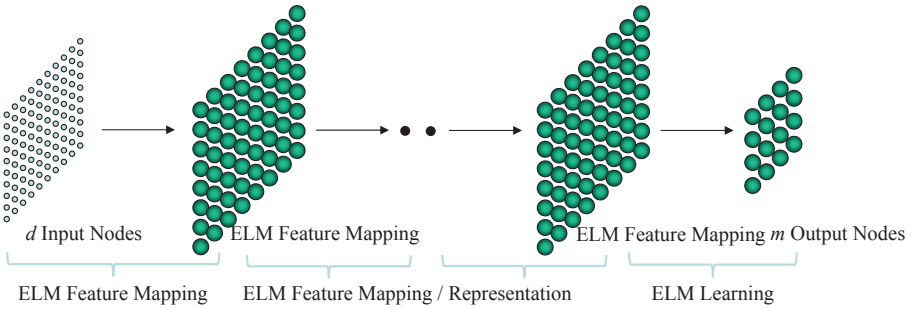
C. Relationship and Differences Among ELM, Deep Learning and SVM/LS-SVM

ELM is different from Deep Learning in the sense that hidden neurons of the entire ELM do not need to be tuned. Due to ELM's different roles of feature learning and clustering, ELM can be used as the earler layers in multi layer networks in which the late layers are trained by other methods such as deep learning (cf. Figure 5).

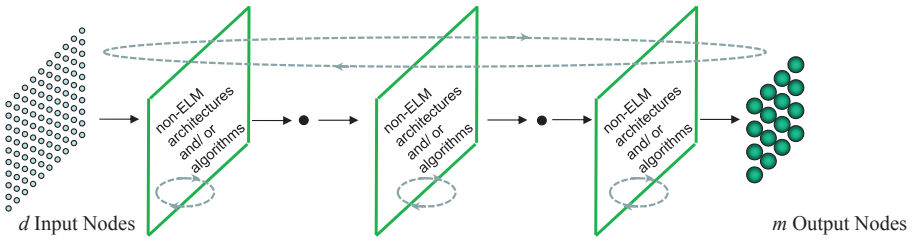
SVM was originally proposed to handle multi layer feedforward networks by Cortes and Vapnik [15] which assumes that when there is no algorithm to train a multi layer network, one can consider the output function of the last hidden layer as $\phi(\mathbf{x})$.

1) Unlike ELM and deep learning which study feature representations in each layer, SVM and LS-SVM do not consider the feature representation and functioning roles of each inner hidden layer (cf. Figure 7).
2) SVM and LS-SVM can also be considered as single hidden layer networks with the hidden layer output function $\phi(\mathbf{x})$. In this case, both ELM and SVM/LS-SVM have single hidden layers. However, ELM has explicit hidden layer mapping $\mathbf{h}(\mathbf{x})$ (convenient for feature representations) and SVM/LS-SVM has unknown hidden layer mapping $\phi(\mathbf{x})$ (inconvenient for feature representations).
3) ELM works for feature learning, clustering regression and classification with ridge regression optimization condition while SVM/LS-SVM mainly works for binary classification with maximal margin optimization condition. It is difficult for SVM / LS-SVM to have feature representation due to unknown mapping $\phi(\mathbf{x})$. (Refer to Table I for detail comparisons between ELM and SVM / LS-SVM, and Huang, *et al.* [6], [23] for detail analysis on the reasons why SVM and LS-SVM provide suboptimal solutions in general.)

[5]http://en.wikipedia.org/wiki/Frank-Rosenblatt

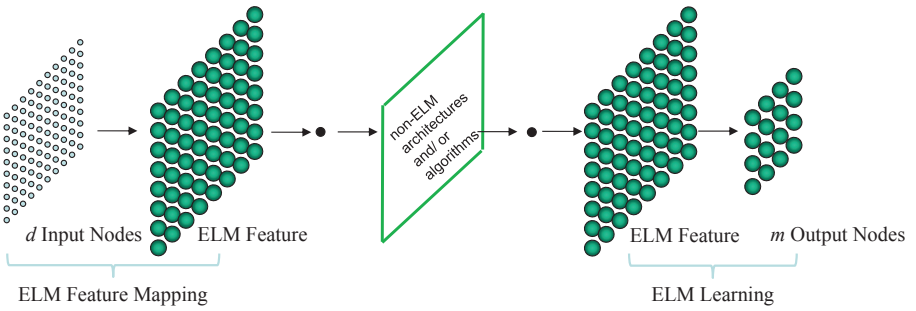


(a) ELM: Entire network as a big single ELM but with ELM working for each layer. Each layer has feature presentation and is trained without hidden neurons tuned. (e.g., [8], [38], [39])

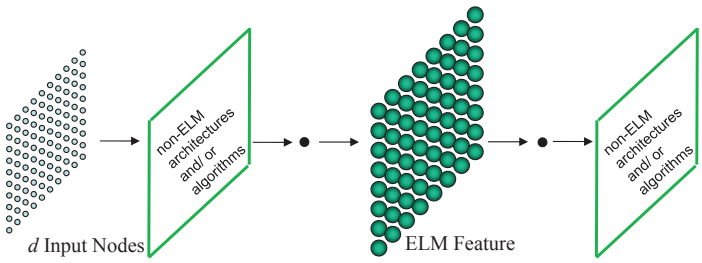


(b) Deep learning: Feature representations are given in hidden layers. Hidden neurons are iteratively tuned in inner models, and iterative tuning are imposed on the entire networks

Fig. 4. Comparisons between hierarchical ELM and deep learning: Each ELM slice forms one hidden layer and hidden node in some hidden layers could be a subnetwork of several neurons. Unlike deep learning concept, ELM (at its both single hidden layer or multi hidden layers of architectures) emphasizes in learning without tuning hidden neurons.



(a) Other learning models work between different ELM slices



(b) ELM slices work between different learning models

Fig. 6. ELM slices work with different learning models: However, each ELM slice as a fundamental learning element can be incorporated into other learning models (e.g., [40]–[44]).

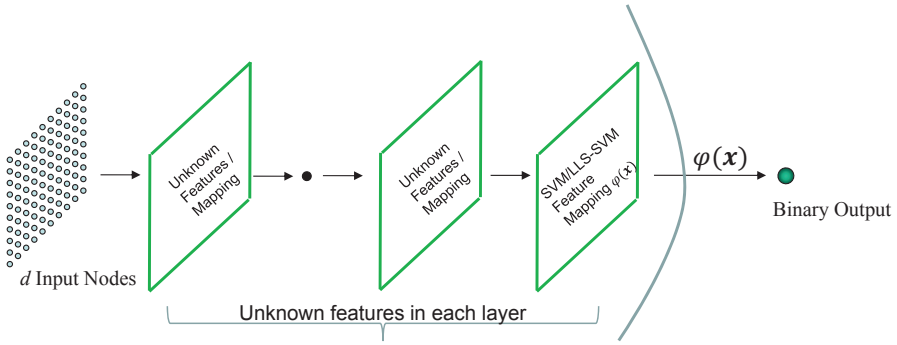


Fig. 7. Relationships and differences between ELM, SVM / LS-SVM and deep learning: Unlike ELM and deep leaning, 1) SVM / LS-SVM as Multi-layers of networks do not emphasize feature representations in hidden layers; and 2) SVM / LS-SVM only handle binary cases directly in their original formula.

ELM is a collection of 3 different algorithms plagiarized from different sources. Which version are you talking about in Table 1? In particular, ELM-Kernel is identical to LS-SVM with b=0. Are you deliberately mixing up the 3 versions to create confusion ?

TABLE I
RELATIONSHIP AND DIFFERENCE COMPARISON BETWEEN ELM AND SVM / LS-SVM

Properties	**ELMs**	**SVM**	**LS-SVM**
Belief	Unlike conventional learning theories and common understanding, ELM belief: Learning can be made without tuning hidden neurons in wide type of biological learning mechanisms and wide types of neural networks	No such belief (Original assumption [15]: If there is no learning solution for feedforward networks, one only needs to consider the output of the last hidden layer: $\phi(\mathbf{x})$)	No such belief (Original assumption [15]: If there is no learning solution for feedforward networks, one only needs to consider the output of the last hidden layer: $\phi(\mathbf{x})$)
Biological inspired	Yes (Confirmed in rats' olfactory system / visual system) This is laughable.	No	No
Network output functions	$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x})$	$f(\mathbf{x}) = \sum_{s=1}^{N_s} \alpha_s t_s \phi(\mathbf{x}) \cdot \phi(\mathbf{x}_s) + b$	$f(\mathbf{x}) = \sum_{s=1}^{N_s} \alpha_s t_s \phi(\mathbf{x}) \cdot \phi(\mathbf{x}_s) + b$
Multiclass classification	Direct solutions	Indirect solutions based on binary ($t_i = 0$ or 1) case	Indirect solutions based on binary ($t_i = 0$ or 1) case
Explicit feature mappings	Yes (Wide types of explicit feature mappings $\mathbf{h}(\mathbf{x})$. Kernels can also be used.)	No (Unknown mapping $\phi(\mathbf{x})$, kernel only.)	No (Unknown mapping $\phi(\mathbf{x})$, kernel only.)
Hidden node types (mathematical model)	Wide types (sigmoid, kernel, Fourier series, etc)	Kernels	Kernels
Hidden node types (biological neurons)	Yes	No	No
Domain	Both real and complex domains	Real domain (Difficult in handling complex domain directly)	Real domain (Difficult in handling complex domain directly)
SLFNs	"Generalized" SLFN Wide types of SLFNs	No	No
Layer wise feature representation	Yes	No (Feature representations in different layers are ignored)	No (Feature representations in different layers are ignored)
Connectivity	For both fully connected and randomly (partially) connected network	No attention on network connections	No attention on network connections
Hyperplane constraints in dual problem	No (It has no such hyperplane constraints due to lack of bias b in output nodes.)	Yes (It has such hyperplane constraints due to bias b in output nodes.)	Yes (It also provides the model without b but it does still assume binary class. [45])
Universal approximation and classification capability	Proved theoretically for wide types of nonlinear piecewise nodes / neurons	No theoretical proof	No theoretical proof
Ridge regression theory	Yes (Consistent for feature learning, clustering, regression and binary / multiclass classification.)	No (Maximal margin concept is a specific case of ridge regression theory used in binary classification.)	No (Maximal margin concept is a specific case of ridge regression theory used in binary classification.)
Learning capability	Efficient in feature learning (auto-encoders) and clustering	Difficult in handling auto-encoders	Difficult in handling auto-encoders
Solutions	Closed-form and non-closed-form, online, sequential and incremental	Non-closed-form	Closed-form

V. Hidden Neuron Types

Unlike Schmidt, *et al.* [1] and Pao, *et al.* [32] in which each node is a sigmoid or RBF function, ELM is valid for wide types of neural nodes and non neural nodes. ELM is efficient for kernel learning as well [6], [23].

You are not the first to use these neuron types. There is nothing stopping Pao or Schmidt from using these different neurons.

A. Real Domain

As ELM has universal approximation capability for a wide type of nonlinear piecewise continuous functions $G(\mathbf{a}, b, \mathbf{x})$, it does not need any bias in the output layer. Some commonly used activation functions covered in ELM theories are:

1) Sigmoid function:

$$G(\mathbf{a}, b, \mathbf{x}) = \frac{1}{1 + \exp(-(\mathbf{a} \cdot \mathbf{x} + b))} \tag{4}$$

2) Fourier function [25], [46]:

$$G(\mathbf{a}, b, \mathbf{x}) = \sin(\mathbf{a} \cdot \mathbf{x} + b) \tag{5}$$

3) Hardlimit function [25], [36]:

$$G(\mathbf{a}, b, \mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{a} \cdot \mathbf{x} - b \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

4) Gaussian function [23], [25]:

$$G(\mathbf{a}, b, \mathbf{x}) = \exp(-b\|\mathbf{x} - \mathbf{a}\|^2) \tag{7}$$

5) Multiquadrics function [23], [25]:

$$G(\mathbf{a}, b, \mathbf{x}) = (\|\mathbf{x} - \mathbf{a}\|^2 + b^2)^{1/2} \tag{8}$$

6) Wavelet [47], [48]:

$$G(\mathbf{a}, b, \mathbf{x}) = \|a\|^{-1/2} \Psi\left(\frac{\mathbf{x} - \mathbf{a}}{b}\right) \tag{9}$$

where Ψ is a single mother wavelet function.

Remark: Due to the validity of universal approximation and classification capability on general nonlinear piecewise continuous activation functions, *combinations of different type of hidden neurons can be used in ELM* [49].

B. Complex Domain

According to Li, et al. [4], [37], randon hidden nodes used in ELM can be fully complex hidden nodes proposed by Kim and Adali [50], and the resultant ELM in complex domain has the universal approximation capability too. The complex hidden nodes of ELM include but are not limited to:

1) Circular functions:

$$\tan(z) = \frac{e^{iz} - e^{-iz}}{i(e^{iz} + e^{-iz})} \tag{10}$$

$$\sin(z) = \frac{e^{iz} - e^{-iz}}{2i} \tag{11}$$

2) Inverse circular functions:

$$\arctan(z) = \int_0^z \frac{dt}{1 + t^2} \tag{12}$$

$$\arccos(z) = \int_0^z \frac{dt}{(1 - t^2)^{1/2}} \tag{13}$$

3) Hyperbolic functions:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{14}$$

$$\sinh(z) = \frac{e^z - e^{-z}}{2} \tag{15}$$

4) Inverse hyperbolic functions:

$$\operatorname{arctanh}(z) = \int_0^z \frac{dt}{1 - t^2} \tag{16}$$

$$\operatorname{arcsinh}(z) = \int_0^z \frac{dt}{(1 + t^2)^{1/2}} \tag{17}$$

VI. Regularization Network and Generalization Performance

Similar to most conventional learning algorithms proposed in 1980s-1990s, Schmidt, *et al.* [1] and Pao, *et al.* [32] focus on minimizing training errors only. They are not regularization networks.[6]

However, inspired by neural networks generalization performance theories proposed in 1998 [14], which are published after Schmidt, *et al.* [1] and Pao, *et al.* [32], ELM theory aims to reach the smallest training error but also the smallest norm of output weights [24], [53] (in this sense, general speaking, ELM is a kind of regularization neural networks but with non-tuned hidden layer mappings (formed by either random hidden nodes, kernels or other implementations)):

$$\text{Minimize: } \|\boldsymbol{\beta}\|_p^{\sigma_1} + C\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_q^{\sigma_2} \tag{18}$$

where $\sigma_1 > 0, \sigma_2 > 0, p, q = 0, \frac{1}{2}, 1, 2, \cdots, +\infty$. *Different combinations of* $\|\boldsymbol{\beta}\|_p^{\sigma_1}$ *and* $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_q^{\sigma_2}$ *can be used and result in different learning algorithms for feature learning and clustering [7].* $\mathbf{H}$ is the ELM hidden layer output matrix (*randomized matrix*):

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_N) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix} \tag{19}$$

and $\mathbf{T}$ is the training data target matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix} = \begin{bmatrix} t_{11} & \cdots & t_{1m} \\ \vdots & \vdots & \vdots \\ t_{N1} & \cdots & t_{Nm} \end{bmatrix} \tag{20}$$

One can linearly apply many ELM solutions (but not all) to the specific sigmoid network with b (Schmidt, *et al.* [1]) and a network with direct link from the input layer to the output network (including but not limited to QuickNet [54] and RVFL [2]), suboptimal solutions will be reached compared to the original ELM. The resultant learning algorithms can

[6]Chen, *et al.* [51], [52] provide some interesting learning algorithms for RVFL networks and suggested that regularization could be used to avoid overfitting. Their works are different from structural risk minimization and maximum margin concept adopted in SVM. ELM's regularization objective moves beyond maximum margin concept and ELM is able to unify neural network generalization theory, structral risk minimization control theory, matrix theory, and linear system theory in ELM learning models. (Refer to Huang [6], [23] for detail analysis.)

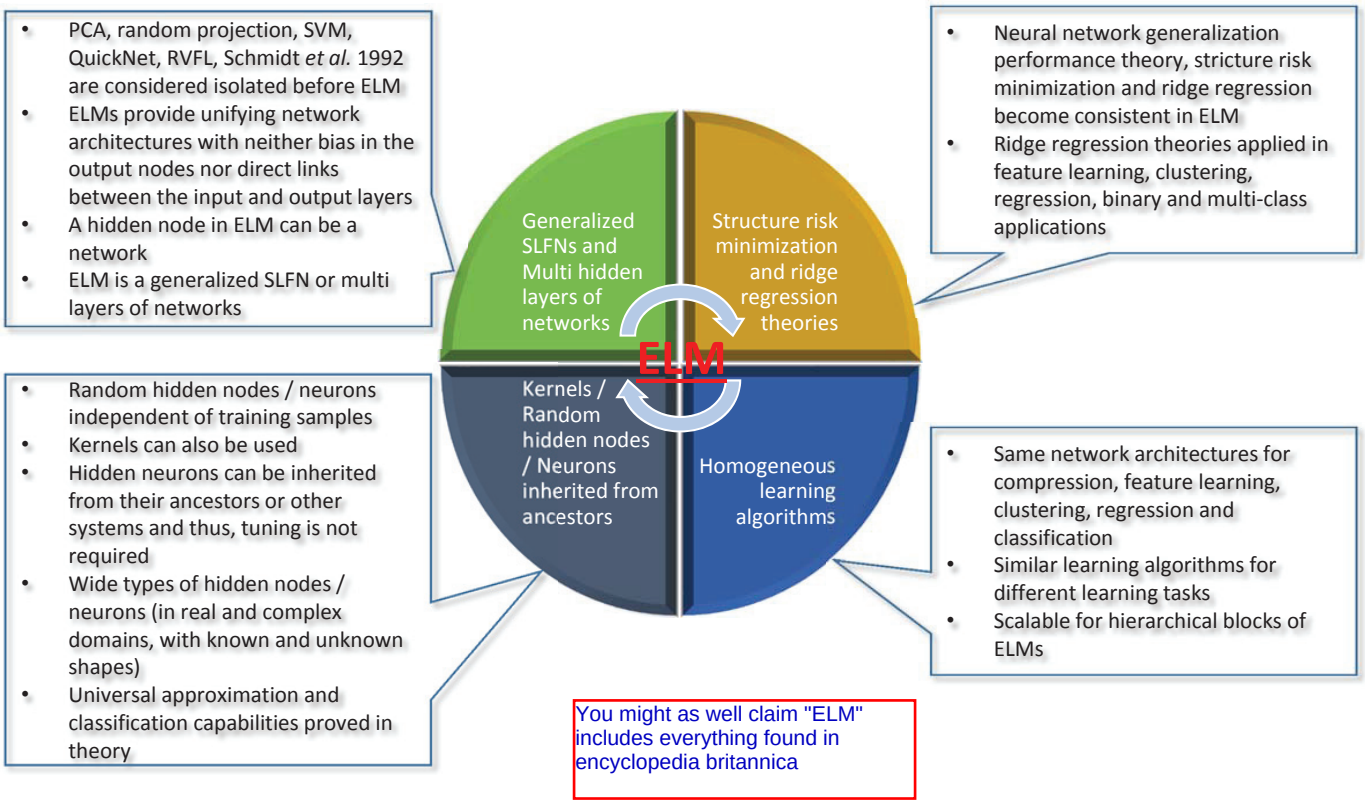


Fig. 8. Essential elements of ELM.

be referred to ELM+b and ELM+$\alpha\mathbf{x}$, respectively. (Refer to Huang, et al. [6], [23] for details.)

For RVFL, the hidden layer output matrix is:

$$\mathbf{H}_{\text{RVFL}} = \begin{bmatrix} g_{\text{sig, RBF}}(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & g_{\text{sig, RBF}}(\mathbf{a}_L, b_L, \mathbf{x}_1) & \mathbf{x}_1 \\ \vdots & \vdots & \vdots & \vdots \\ g_{\text{sig, RBF}}(\mathbf{a}_1, b_1, \mathbf{x}_N) & \cdots & g_{\text{sig, RBF}}(\mathbf{a}_L, b_L, \mathbf{x}_N) & \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{\text{ELM for sigmoid or RBF basis}} & \mathbf{X}_{N\times d} \end{bmatrix} \tag{21}$$

where $\mathbf{H}_{\text{ELM for sigmoid or RBF basis}}$ are two specific ELM hidden layer output matrices (19) with sigmoid or RBF basis, and $\mathbf{X}_{N\times d}$ is a $N \times d$ matrix with i-th input $\mathbf{x}_i$ as the i-th row.

If the output neuron bias is considered as a bias neuron in the hidden layer as done in most conventional neural networks, the hidden layer output matrix for Schmidt, *et al.* [1] will be

$$\mathbf{H}_{\text{Schmidt, } et\ al.(1992)} = \begin{bmatrix} g_{\text{sig}}(\mathbf{a}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g_{\text{sig}}(\mathbf{a}_L \cdot \mathbf{x}_1 + b_L) & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & & \vdots & \\ g_{\text{sig}}(\mathbf{a}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g_{\text{sig}}(\mathbf{a}_L \cdot \mathbf{x}_N + b_L) & 1 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{\text{ELM for sigmoid basis}} & \mathbf{1}_{N\times m} \end{bmatrix} \tag{22}$$

where $\mathbf{H}_{\text{ELM for sigmoid basis}}$ is a specific ELM hidden layer output matrix (19) with sigmoid basis, and $\mathbf{1}_{N\times m}$ is a $N \times m$ matrix with constant element 1. Although bias b in Schmidt, *et al.* [1] seems like a simple parameter, however, it is known that from both mathematical and machine learning point of view, a parameter could result in some significant differences. Its role has drawn researchers' attention [6], [55], [56]. In fact, one of the main reasons why it is difficult to apply SVM and LS-SVM in multiclass applications in the past two decades is mainly due to the output node bias b. Without the output node bias b SVM and LS-SVM solutions will become much easier [6], [23]. Without bias, these are known as kernel ridge regression since 1998. Again, copied concepts from the literature.

VII. Closed-Form Solutions vs Non-Closed-Form Solutions

In many cases closed form solutions of ELM can be given when (18) $\sigma_1 = \sigma_2 = p = q = 2$. However, non-closed-form solutions can be given if $\sigma_1 = \sigma_2 = p = q = 2$ [5], [13], [25], [26], [57] or if other values are given to σ_1,σ_2, p, and q especially when ELM is used in the applications of compression, feature learning and clustering [5], [7], [25], [26], [58]–[60]. Actually the original proof on the universal approximation capability of ELM is based on non-closed form solutions of ELM [5], [25], [26].

VIII. Conclusions

It has been around 60 years since Frank Rosenblatt [9] dreamed that his perceptron could enable computers to "walk, talk, see, write, reproduce itself and be conscious to its existence." It was difficult for many researchers to believe his great dream due to lack of efficient learning algorithms and strong theoretical support in the very beginning of artificial neural network era. On the other hand, John von Neumann was puzzled [19], [20] why "an imperfect neural network,

Huang uses words/phrases such as analyzed, discussed, sub-optimal, point of view, etc. to make derogatory remarks about original methods like SVM, LS-VM, RVFL, RBF, etc and to make favourable comments about the unethical name ELM which is minute variant of the original methods. Extensive simulation results in JMLR and in DOI: 10.1016/j.ins.2015.09.025 show just the opposite.

containing many random connections, can be made to perform reliably those functions which might be represented by idealized wiring diagrams" [9]. This paper shows that ELM theories and framework may fill such a gap between Frank Rosenblatt's dream and John von Neumann's puzzle:

1) ELM can be used to train wide type of multi hidden layer of feedforward networks: Each hidden layer can be trained by one single ELM based on its role as feature learning, clustering, regression or classification. Entire network as a whole can be considered as a single ELM in which hidden neurons need not be tuned. (Refer to Figure 8 for the detail summary of ELM.)
2) ELM slice can be "inserted" into many local parts of a multi hidden layer feedforward network, or work together with other learning architectures / models.
3) A hidden node in an ELM slice (a "generalized" SLFN) can be a network of several nodes, thus local receptive fields can be formed.
4) In each hidden layer, input layers to hidden nodes can be fully or partially randomly connected according to different continuous probability distribution function. The performance of the network is stable even if a finite number of hidden neurons and their related connections change.

Thus, overall speaking, from ELM theories point of view, the entire multi layers of networks are structured and ordered, but they may be seemingly "messy" and "unstructured" in a particular layer or neuron slice. "Hard wiring" can be randomly built locally with full connection or partial connections. Co-existence of globally structured architectures and locally random hidden neurons happen to have fundamental learning capabilities of compression, feature learning, clustering, regression and classification. This may have addressed John von Neumann's puzzle. Biological learning mechanisms are sophisticated, we believe that "learning without tuning hidden neurons" is one of fundamental biological learning mechanisms in many modules of learning systems. Furthermore, random hidden neurons and "random wiring" are only two specific implementations of such "learning without tuning hidden neurons" learning mechanisms.

This author knowingly excluded very similar RBF papers from his ELM-RBF articles in 2004 in order to mislead the reviewers.

ACKNOWLEDGMENT

Minsky and Rosenblatt's controversy may have indirectly inspired the reviving of artificial neural network research in 1980s. Their controversy turned out to show that hidden neurons are critical. Although the main stream of research have focused learning algorithms on tuning hidden neurons since 1980s, few pioneers independently studied alternative solutions (e.g., Halbert White for QuickNet, Yao-Han Pao, Boris Igelnik, and C. L. Philip Chen for RVFL, D. S. Broomhead and David Lowe for RBF networks, Corinna Cortes and Vladimir Vapnik for SVM, J. A. K. Suykens and J. Vandewalle for LS-SVM, Baum, Wouter F. Schmidt, Martin A. Kraaijveld and Robert P. W. Duin for Random Weights Sigmoid Network) in 1980s-1990s. Although few of those attempts did not take off finally due to various reasons and constrains, they have been playing significant and irreplaceable roles in the relevant research history. We would appreciate their historical contributions. As discussed in Huang [6], without BP and SVM / LS-SVM, the research and applications on neural networks would never have been so intensive and extensive.

We would like to thank Bernard Widrow, Stanford University, USA for sharing his vision in neural networks and precious historical experiences with us in the past years. It is always pleasant to discuss with him on biological learning, neuroscience, and Rosenblatt's work. We both feel confident that we have never been so close to natural biological learning.

We would like to thank C. L. Philip Chen, University of Macau, China and Boris Igelnik, BMI Research, Inc., USA for invaluable discussions on RVFL, Johan Suykens, Katholieke Universiteit Leuven, Belgium for invaluable discussions on LS-SVM, Stefano Fusi, Columbia University, USA for the discussion on the links between biological learning and ELM.

We would also like to thank Jose Principe, University of Florida, USA for nice discussions on neuroscience (especially on neuron layers / slices) and his invaluable suggestions on potential mathematical problems of ELM, and M. Brandon Westover, Harvard Medical School, USA for the constructive comments and suggestions on the potential links between ELM and biological learning in local receptive fields.

Citing incorrectly is worse than not citing at all. You MUST explain and compare correctly. You make confusing statements repeatedly. You have NEVER stated the real trivial difference between RVFL and ELM.

APPENDIX: FURTHER CLARIFICATION OF MISUNDERSTANDINGS ON THE RELATIONSHIP BETWEEN ELM AND SOME EARLIER WORKS

Recently, it has drawn our attention that several researchers have been making some very negative and unhelpful comments on ELM in neither academic nor professional manner due to various reasons and intentions, which mainly state that ELM does not refer to some earlier works (e.g., Schmidt, *et al.* (1992) and RVFL), and ELM is the same as those earlier works. It should be pointed out that these earlier works have actually been referred in our different publications on ELM as early as in 2007. It is worth mentioning that ELM actually provides a unifying learning platform for wide types of neural networks and machine learning techniques by absorbing the common advantages of many seemingly isolated different research efforts made in the past 60 years, and thus, it may not be surprising to see apparent relationships between ELM and different techniques. As analyzed in this paper, the essential differences between ELM and those mentioned related works are subtle but crucial.

Trivial, but makes ELM performs worse than RVFL.

It is not rare to meet some cases in which intuitively speaking some techniques superficially seem similar to each other but actually they are significantly different. ELM theories provide a unifying platform for wide types of neural networks, Fourier series [25], wavelets [47], [48], mathematical series [5], [25], [26], etc. Although the relationship and differences between ELM and those earlier works have clearly been discussed in the main context of this paper, in response to the anonymous malign letter, some more background and discussions need to be highlighted in this appendix further.

A. Misunderstanding on References

Several researchers thought that ELM community has not referred to those earlier related work, e.g., Schmidt, *et al.* [1], RVFL [2], and Broomhead and Lowe [61]. We wish to draw their serious attention that our earlier work (2008) [4] has explicitly stated: "Several researchers [31], [32], [35], [62] (e.g., Baum (1988), Broomhead and Lowe (1988), Igelnik and Pao (1995), Ferrari and Stengel (2005)) have independently found that the input weights or centers $\mathbf{a}_i$ do not need to be tuned." (These works were published in different years, one did not refer to the others. Ferrari and Stengel (2005) has been kindly referred in our work although it was even published later than ELM [24].) In addition, in contrast to the misunderstanding that those earlier works were not referred, we have even referred to Baum (1988) [31]'s work and White's QuickNet (1989) [54] in our 2008 works [4], [5], which we consider much earlier than Schmidt, *et al.* (1992) [1] and RVFL (1994) [2] in the related research areas. There is also a misunderstanding that Park and Sandberg's RBF theory [21] has not been referred in ELM work. In fact, Park and Sandberg's RBF theory has been referred in the proof of ELM's theories on RBF cases as early as 2006 [25].

Although we did not know Schmidt, *et al.* [1] until 2012, we have referred to it in our work [6] immediately. We spent almost 10 years (back to 1996) on proving ELM theories and may have missed some related works. However, from literature survey point of view, Baum [31] may be the earliest work we could find so far and has been referred at the first time. Although Schmidt, *et al.* [1] is interesting, the citations of Schmidt, *et al.* [1] was almost zero before 2013 (Google Scholar), it is not easy for his work to turn up in search engine unless one intentionally flips hundreds of search pages. Such information may not be available in earlier generation of search engine when ELM was proposed. The old search engines available in the beginning of this century were not as powerful as most available search engines and many publications were not online 10-15 years ago. As stated in our earlier work [4], Baum [31] claimed that (seen from simulations) one may fix the weights of the connections on one level and simply adjust the connections on the other level and no (significant) gain is possible by using an algorithm able to adjust the weights on both levels simultaneously. Surely, almost every researcher knows that the easiest way is to calculate the output weights by least-square method (close-form) as done in Schmidt, *et al.* [1] and ELM if the input weights are fixed.

B. Loss of Feature Learning Capability

The earlier works (Schmidt, *et al.* [1], RVFL [2], Broomhead and Lowe [61]) may lose learning capability in some cases.

As analyzed in our earlier work [3], [4], although Lowe [35]'s RBF network chooses RBF network centers randomly, it uses one value b for all the impact factors in all RBF hidden nodes, such a network will lose learning capability if the impact factor b is randomly generated. Thus, in RBF network implementation, the single value of impact factors is usually adjusted manually or based on cross-validation. In this sense, Lower's RBF network does not use random RBF hidden neurons, let alone wide types of ELM networks.[7] Furthermore, Chen, *et al.* [64] points out that such Lowe [35], [61]'s RBF learning procedure may not be satisfactory and thus they proposed an alternative learning procedure to choose RBF node centers one by one in a rational way which is also different from random hidden nodes used by ELM.

Schmidt, *et al.* [1] at its original form may face difficulty in sparse data applications, however, one can linearly extend sparse ELM solutions to Schmidt, *et al.* [1] (the resultant solution referred to ELM+b).

ELM is efficient for auto encoder as well [39]. However, when RVFL is used for auto encoder, the weights of the direct link between its input layer to its output layer will become a constant value one and the weights of the links between its hidden layer to its output layer will become a constant value zero, thus, RVFL will lose learning capability in auto encoder cases. Schmidt, *et al.* [1] which has the biases in output nodes may face difficulty in auto encoder cases too.

It may be difficult to implement those earlier works (Schmidt, *et al.* [1], RVFL [2], Broomhead and Lowe [61]) in multi layer networks, while hierarchical ELM with multi ELM each working in one hidden layer can be considered as a single ELM itself. Table II summarizes the relationship and main differences between ELM and those earlier works.

REFERENCES

[1] W. F. Schmidt, M. A. Kraaijveld, and R. P. W. Duin, "Feed forward neural networks with random weights," in *Proceedings of 11th IAPR International Conference on Pattern Recognition Methodology and Systems*, (Hague, Netherlands), pp. 1–4, 1992.

[2] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, pp. 163–180, 1994.

[3] G.-B. Huang, "Reply to "comments on 'the extreme learning machine' "," *IEEE Transactions on Neural Networks*, vol. 19, no. 8, pp. 1495–1496, 2008.

[4] G.-B. Huang, M.-B. Li, L. Chen, and C.-K. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, pp. 576–583, 2008.

[5] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, pp. 3460–3468, 2008.

[6] G.-B. Huang, "An insight into extreme learning machines: Random neurons, random features and kernels," *Cognitive Computation*, vol. 6, no. 3, pp. 376–390, 2014.

[7] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2405–2417, 2014.

[8] G.-B. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong, "Local receptive fields based extreme learning machine," *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 18–29, 2015.

[9] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[10] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proceedings of the 2007 Neural Information Processing Systems (NIPS2007)*, pp. 1177–1184, 3-6 Dec, 2007.

[7] Differences between Lowe's RBF networks and ELM have been clearly given in our earlier reply [3] in response to another earlier comment letter on ELM [63]. It is not clear why several researchers in their anonymous letter refer to the comment letter on ELM [63] for Lowe [35]'s RBF network and RVFL but do not give readers right and clear information by referring to the response [3].

This table is misleading and non-sense. All so called extensions of the ELM could have been done better using the RVFL. The author must explain why RVFL published in 1994 and Randomized NN published in 1992 can't be extended with or without trivial modifications to them.

Hence, there is no justification for creating the ELM name unethically in your IJCNN 2004 and ICARCV 2004 papers.

TABLE II

RELATIONSHIP AND DIFFERENCE COMPARISON AMONG DIFFERENT METHODS: ELMS, SCHMIDT, *et al.* (1992), QUICKNET / RVFL

Properties	**ELMs**	**Schmidt, *et al.* (1992)**	**QuickNet / RVFL**
Belief (Are we talking about religion now ?)	Unlike conventional learning theories and common understanding, ELM belief: Learning can be made without tuning hidden neurons in wide type of biological learning mechanisms and wide types of neural networks	No such belief	No such belief
Network output functions	$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x})$	$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i g_{\text{sig}}(\mathbf{a}_i \cdot \mathbf{x} + b_i) + b$	$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i g_{\text{sig or RBF}} + \alpha \cdot \mathbf{x}$
SLFNs	"Generalized" SLFN in which a hidden node can be a subnetwork	Standard SLFN only	Standard SLFN only
Multi layers networks	Yes	No	No
Connectivity	For both fully connected and randomly (partially) connected network	Fully connected	Fully connected
Hidden node types (mathematical model)	Wide types (sigmoid, kernel, Fourier series, etc)	Sigmoid	Sigmoid and RBF
Hidden node types (biological neurons)	Yes	No	No
Domain	Both real and complex domains	Real domain	Real domain
Hidden layer output matrix	$\mathbf{H}_{\text{ELM}}$	$[\mathbf{H}_{\text{ELM for sigmoid basis}}, \mathbf{1}_{N\times m}]$	$[\mathbf{H}_{\text{ELM for sigmoid or RBF basis}}, \mathbf{X}_{N\times d}]$
Universal approximation and classification capability	Proved for wide types of random neurons	No theoretical proof	Theoretical proof for semi-random sigmoid or RBF nodes
Structural risk minimization	Minimize: $\|\boldsymbol{\beta}\|_p^{\sigma_1} + C\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_q^{\sigma_2}$	Not considered	Not considered
Learning capability	Efficient in feature learning (auto-encoders) and clustering	Difficult in handling auto-encoders	Lose learning capability in auto-encoders
Solutions	Closed-form and non-closed-form, sequential and incremental	Closed-form	Closed-form and non-closed-form for QuickNet, Closed-form for RVFL
Portability	Many (*but not all*) ELM variants can be linearly extended to Schmidt, *et al.* (1992) and RVFL/QuickNet instead of vice versa, the resultant algorithms are referred to as "ELMs+b" for Schmidt, *et al.* (1992) and "ELM+$\alpha\mathbf{x}$" for QuickNet/RVFL		

ELM is a collection of 3 different algorithms plagiarized from different sources. Which version are you talking about in Table 1? You deliberately mix up the 3 versions of ELM to create confusion. In particular, ELM-Kernel is identical to LS-SVM with b=0 and kernel ridge regression. If you compare ELM-Kernel with Schmidt/RVFL, of course they are different! You should compare ELM-SLFN with Schmidt/RVFL, then the difference is tiny: tunable bias in Schmidt/RVFL but zero bias in ELM-SLFN (non-optimal).

pansions in logical time," in *Proceedings of the 30th International Conference on Machine Learning*, (Atlanta, USA), 16-21 June 2013.

[12] P.-S. Huang, L. Deng, M. Hasegawa-Johnson, and X. He, "Random features for kernel deep convex network," in *Proceedings of The 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, (Vancouver, Canada), 26-31 May 2013.

[13] B. Widrow, A. Greenblatt, Y. Kim, and D. Park, "The no-prop algorithm: A new learning algorithm for multilayer neural networks," *Neural Networks*, vol. 37, pp. 182–188, 2013.

[14] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.

[15] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[16] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[17] M. Minski and S. Papert, "Perceptrons: An introduction to computational geometry," MIT Press, 1969.

[18] G.-B. Huang, "Learning capability of neural networks," (Ph.D. thesis, Nanyang Technological University, Singapore), 1998.

[19] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata studies* (C. E. Shannon and J. McCarthy, eds.), pp. 43–98, Princeton: Princeton University Press, 1956.

[20] J. von Neumann, "The general and logical theory of automata," in *Cerebral mechanisms in behavior* (L. A. Jeffress, ed.), pp. 1–41, New York: Wiley, 1951.

[21] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, pp. 246–257, 1991.

[22] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, pp. 861–867, 1993.

[23] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 42, no. 2, pp. 513–529, 2012.

[24] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proceedings* vol. 2, (Budapest, Hungary), pp. 985–990, 25-29 July, 2004.

[25] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[26] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, pp. 3056–3062, 2007.

[27] D. L. Sosulski, M. L. Bloom, T. Cutforth, R. Axel, and S. R. Datta, "Distinct representations of olfactory information in different cortical centres," *Nature*, vol. 472, pp. 213–216, 2011.

[28] C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen, "A large-scale model of the functioning brain," *Science*, vol. 338, pp. 1202–1205, 2012.

[29] O. Barak, M. Rigotti, and S. Fusi, "The sparseness of mixed selectivity neurons controls the generalizationcdiscrimination trade-off," *The Journal of Neuroscience*, vol. 33, no. 9, pp. 3844–3856, 2013.

[30] M. Rigotti, O. Barak, M. R. Warden, X.-J. Wang, N. D. Daw, E. K. Miller, and S. Fusi, "The importance of mixed selectivity in complex cognitive tasks," *Nature*, vol. 497, pp. 585–590, 2013.

[31] E. Baum, "On the capabilities of multilayer perceptrons," *Journal of Complexity*, vol. 4, pp. 193–215, 1988.

[32] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1320–1329, 1995.

[33] S. Tamura and M. Tateishi, "Capabilities of a four-layered feedforward neural network: Four layers versus three," *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 251–255, 1997.

[34] J. Principle and B. Chen, "Universal approximation with convex optimization: Gimmick or reality?," *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 68–77, 2015.

[35] D. Lowe, "Adaptive radial basis function nonlinearities and the problem of generalisation," in *Proceedings of First IEE International Conference on Artificial Neural Networks*, pp. 171–175, 1989.

[36] G.-B. Huang, Q.-Y. Zhu, K. Z. Mao, C.-K. Siew, P. Saratchandran, and N. Sundararajan, "Can threshold networks be trained directly?," *IEEE Transactions on Circuits and Systems II*, vol. 53, no. 3, pp. 187–191, 2006.

[37] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, pp. 306–314, 2005.

[38] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine

for multilayer perceptron," *(DOI:10.1109/TNNLS.2015.2424995) IEEE Transactions on Neural Networks and Learning Systems*, 2015.

[39] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational learning with extreme learning machine for big data," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 31–34, 2013.

[40] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition," in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*, (Kyoto, Japan), September 29-October 2 2009.

[41] A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng, "On random weights and unsupervised feature learning," in *Proceedings of the 28th International Conference on Machine Learning*, (Bellevue, USA), 28 June-2 July 2011.

[42] D. Cox and N. Pinto, "Beyond simple features: A large-scale feature search approach to unconstrained face recognition," in *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*, pp. 8–15, IEEE, 2011.

[43] M. D. McDonnell and T. Vladusich, "Enhanced image classification with a fast-learning shallow convolutional neural network," in *Proceedings of International Joint Conference on Neural Networks (IJCNN'2015)*, (Killarney, Ireland), July 12-17, 2015.

[44] Y. Zeng, X. Xu, Y. Fang, and K. Zhao, "Traffic sign recognition using extreme learning classifier with deep convolutional features," in *The 2015 International Conference on Intelligence Science and Big Data Engineering (IScIDE 2015)*, (Suzhou, China), June 14-16, 2015.

[45] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, "Least squares support vector machines," Singapore: World Scientific, 2002.

[46] A. Rahimi and B. Recht, "Uniform approximation of functions with random bases," in *Proceedings of the 2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 555–561, 23-26 Sept, 2008.

[47] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, pp. 909–996, 1988.

[48] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[49] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: Optimally pruned extreme learning machine," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 158–162, 2010.

[50] T. Kim and T. Adali, "Approximation by fully complex multilayer perseptrons," *Neural Computation*, vol. 15, pp. 1641–1666, 2003.

[51] C. L. P. Chen, "A rapid supervised learning neural network for function interpolation and approximation," *IEEE Transactions on Neural Networks*, vol. 7, no. 5, pp. 1220–1230, 1996.

[52] C. L. P. Chen and J. Z. Wan, "A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the applications to time-series prediction," *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 29, no. 1, pp. 62–72, 1999.

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

[54] H. White, "An additional hidden unit test for neglected nonlinearity in multilayer feedforward networks," in *Proceedings of the International Conference on Neural Networks*, pp. 451–455, 1989.

[55] T. Poggio, S. Mukherjee, R. Rifkin, A. Rakhlin, and A. Verri, "*b*," (A.I. Memo No. 2001-011, CBCL Memo 198, Artificial Intelligence Laboratory, Massachusetts Institute of Technology), 2001.

[56] I. Steinwart, D. Hush, and C. Scovel, "Training SVMs without offset," *Journal of Machine Learning Research*, vol. 12, no. 1, pp. 141–202, 2011.

[57] J. Luo, C.-M. Vong, and P.-K. Wong, "Sparse bayesian extreme learning machine for multi-classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 4, pp. 836–843, 2014.

[58] S. Decherchi, P. Gastaldo, A. Leoncini, and R. Zunino, "Efficient digital implementation of extreme learning machines for classification," *IEEE Transactions on Circuits and Systems - II*, vol. 59, no. 8, pp. 496–500, 2012.

[59] Z. Bai, G.-B. Huang, D. Wang, H. Wang, and M. B. Westover, "Sparse extreme learning machine for classification," *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1858–1870, 2014.

[60] B. Frénay, M. van Heeswijk, Y. Miche, M. Verleysen, and A. Lendasse, "Feature selection for nonlinear models with extreme learning machines," *Neurocomputing*, vol. 102, pp. 111–124, 2013.

[61] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[62] S. Ferrari and R. F. Stengel, "Smooth function approximation using neural networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 24–38, 2005.

[63] L. P. Wang and C. R. Wan, "Comments on 'the extreme learning machine'," *IEEE Transactions on Neural Networks*, vol. 19, no. 8, pp. 1494–1495, 2008.

[64] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.