An iterative learning algorithm for feedforward neural networks with random weights

Feilong Cao[1], Dianhui Wang[2,3,*], Houying Zhu[4], Yuguang Wang[4]

[1]Department of Applied Mathematics, China Jiliang University,
Hangzhou 310018, Zhejiang, China
[2]Department of Computer Science and Information Technology,
La Trobe University, Melbourne, VIC 3086, Australia.
[3]State Key Lab. of Synthetical Automation for Process Industries,
Northeastern University, Shenyang 110819, Liaoning, China.
[4]School of Mathematics and Statistics,
The University of New South Wales, Sydney 2052, Australia

Abstract

Feedforward neural networks with random weights (FNNRWs), as random basis function approximators, have received considerable attention due to their potential applications in dealing with large scale datasets. Special characteristics of such a learner model come from weights specification, that is, the input weights and biases are randomly assigned and the output weights can be analytically evaluated by a Moore-Penrose generalized inverse of the hidden output matrix. When the size of data samples becomes very large, such a learning scheme is infeasible for problem solving. This paper aims to develop an iterative solution for training FNNRWs with large scale datasets, where a regularization model is employed to potentially produce a learner model with improved generalization capability. Theoretical results on the convergence and stability of the proposed learning algorithm are established. Experiments on some UCI benchmark datasets and a face recognition dataset are carried out, and the results and comparisons indicate the applicability and effectiveness of our proposed learning algorithm for dealing with large scale datasets.

[*]Corresponding author. Emails: icteam@163.com (Feilong Cao), dh.wang@latrobe.edu.au (Dianhui Wang), zhuhouying87@gmail.com (Houying Zhu), wangyg85@gmail.com (Yuguang Wang)

Keywords: Neural networks with random weights, learning algorithm, stability, convergence

1 Introduction

Feedforward neural networks with random weights (FNNRWs) were proposed by Schmidt and his co-workers in [28], which can be mathematically described by

$$G_N(\mathbf{x}) = \sum_{i=1}^{L} \beta_i g\left(\langle \omega_i, \mathbf{x}\rangle + b_i\right), \tag{1.1}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_d]^\intercal \in \mathbf{R}^d$, g is an activation function, b_i is a bias, $\omega_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{id}] \in \mathbf{R}^d$ and $\beta_i \in \mathbf{R}$ are the input and output weights, respectively; $\langle \omega_i, x\rangle = \sum_{j=1}^{d} \omega_{ij} x_j$ denotes the Euclidean inner product.

The input weights and biases are assigned randomly with uniform distribution in $[-1, 1]$, and the output weights can be determined analytically by using the well-known least mean squares method [28]. Unfortunately, such a randomized learner model in [28] was not proposed as a working algorithm, but simply as a tool to investigate some characteristics of feedforward neural networks. From approximation theory, it is obvious that such a way to randomly assign the input weights and biases cannot guarantee the universal approximation capability (in the sense of probability one) of the resulting random learner models. This disability can be verified by many counter examples. Thus, statements on its approximation capability and good generalization in the literature are all misleading and lack scientific justification. A similar randomized learner model, termed as random vector functional-link nets (RVFLs), and associated algorithms were proposed by Pao and his co-workers in [20], where a direct link from the input layer to the output layer is added. In [16], a theoretical justification of the universal approximation capability of RVFLs (without direct link case) was established, where the scope of randomly assigned input weights and biases are data dependant and specified in a constructive manner. Therefore, a careful estimation of the parameters characterizing the scope in algorithm implementation should be done for each dataset in practice.

Due to some good learning characteristics [21], this type of randomized training scheme for feedforward neural networks has been widely applied in data modeling [11].

Obviously, the most appealing property of randomized techniques for feedforward neural networks lies in the possibility of handling large scale datasets with real-time requirement. Recently, some advanced randomized learning algorithms have been developed in [1], [7], [8], [9],[27]. Also, some theoretical results on the approximation power of randomized learners were investigated in [14], [24], [25]. It has been recognized that the way the output weights are computed can result in memory problems for desktop computers as the size of data samples and the number of nodes at the hidden layer of neural networks become very large. Motivated by this crucial drawback and potential applications of such a machine learning technique for big data, we aim to overcome the difficulty of computing the generalized inverse in least mean squares method, and present a feasible solution for the large scale datasets. This work is built on a framework for resolving the linear inverse problems in [10]. Note that our objective in this study is not to investigate properties of the randomized learner model from statistical learning theory, but an iterative scheme for building an instance learner model. It has been recognized that some machine learning techniques, such as ensemble learning [1] and sequential learning [17, 22], are related to large scale data modeling problems, however, our focus in this work is neither on sampling-based ensemble techniques nor streaming data modeling.

The remainder of this paper is organized as follows. Section 2 provides some supportive results for algorithm development. Section 3 proposes an iterative learning algorithm with analyses of its convergence and stability. The performance evaluation is presented in Section 4, where the experimental results on some benchmark datasets and a face recognition dataset are reported. Section 5 concludes this paper with some remarks. Finally, mathematical proofs of the theoretical results are given in the Appendix.

2 Supportive Results

Given a set of training samples $\mathscr{T} = \{(\mathbf{x}_i, t_i) : i = 1, 2, \ldots, N\}$, let ω_i and b_i be chosen randomly from the uniform distribution and fixed in advance. Then, the output weights can be obtained by solving the following linear equation system:

$$\mathbf{H}\beta = \mathbf{T}, \tag{2.1}$$

where $\beta = [\beta_1, \beta_2, \dots, \beta_L]^\intercal$, $\mathbf{T} = [t_1, t_2, \dots, t_N]^\intercal$,

$$\mathbf{H} = \begin{bmatrix} g(\langle \omega_1, \mathbf{x}_1 \rangle + b_1) & \dots & g(\langle \omega_L, \mathbf{x}_1 \rangle + b_L) \\ \vdots & \dots & \vdots \\ g(\langle \omega_1, \mathbf{x}_N \rangle + b_1) & \dots & g(\langle \omega_L, \mathbf{x}_N \rangle + b_L) \end{bmatrix}. \tag{2.2}$$

A least mean squares solution of (2.1) can be expressed by $\beta = \mathbf{H}^\dagger \mathbf{T}$, where $\mathbf{H}^\dagger$ is the Moore-Penrose generalized inverse of $\mathbf{H}$. However, the least squares problem is usually ill-posed and one can employ a regularization model to find a solution, that is,

$$L_\mu(\beta) = \|\mathbf{H}\beta - \mathbf{T}\|_2^2 + \mu J(\beta), \tag{2.3}$$

where μ is a small positive number called the regularizing factor, and J is a differentiable, strictly convex and coercive function with respect to β. In this case, the minimization of (2.3) has a unique solution. Particularly, if we take $J(\beta) = \|\beta\|_2^2$, then the regularization model becomes

$$\Phi_\mu(\beta) = \|\mathbf{H}\beta - \mathbf{T}\|_2^2 + \mu\|\beta\|_2^2. \tag{2.4}$$

If μ is given such that $(\mathbf{H}^\intercal\mathbf{H} + \mu\mathbf{I})$ is invertible, then the minimizer of (2.4) can be written as

$$\beta^\star = (\mathbf{H}^\intercal\mathbf{H} + \mu\mathbf{I})^{-1}\mathbf{H}^\intercal\mathbf{T}, \tag{2.5}$$

where $\mathbf{I}$ denotes the identity matrix.

The regularizing factor is fixed beforehand, and should be taken properly so that $\mathbf{H}^\intercal\mathbf{H} + \mu\mathbf{I}$ continues to be invertible for all random assignments. Furthermore, the computation of the inverse matrix in (2.5) will become very difficult and impractical for large scale datasets. In such a background, it is necessary to develop more effective schemes for problem solving. Based on the iterative method proposed in [10] which is further explored in [31], we first establish the following supportive result (its proof is detailed in Appendix).

Theorem 2.1. *Let $L \geq 1$ and the $\mathbf{R}^L$-valued sequence $\{\beta_k\}_{k=0}^{\infty}$ be defined iteratively as*

$$\beta_k := \frac{1}{M^2+\mu}\left((M^2\mathbf{I} - \mathbf{H}^{\intercal}\mathbf{H})\beta_{k-1} + \mathbf{H}^{\intercal}\mathbf{T}\right),$$

for $k = 1, 2, 3, \ldots$ and $\mu > 0$ is a sufficiently small number. If there exists a positive constant M such that the norm of the output matrix $\mathbf{H}$ given in (2.2) *of the hidden layer is bounded by M, i.e., $\|\mathbf{H}\| < M$, then $\{\beta_k\}_{k=0}^{\infty}$ strongly converges. Further, $\{\beta_k\}_{k=0}^{\infty}$ strongly converges to the minimizer of a regularization model, $\|\mathbf{H}\beta - \mathbf{T}\|_2^2 + \mu\|\beta\|_2^2$, regardless of the choice of β_0.*

Remark 2.1. *From the proof of Theorem 2.1 (see Appendix), we show that for any positive integer q*

$$\|\beta_{k+q} - \beta_k\|_2 \leq \left(\frac{\|M^2\mathbf{I} - \mathbf{H}^{\intercal}\mathbf{H}\|}{M^2+\mu}\right)^k \frac{M^2+\mu}{\mu}\|\beta_1 - \beta_0\|_2.$$

On the left-hand side of the inequality, we let $q \to \infty$, and see that

$$\|\beta^{\star} - \beta_k\|_2 \leq \left(\frac{\|M^2\mathbf{I} - \mathbf{H}^{\intercal}\mathbf{H}\|}{M^2+\mu}\right)^k \frac{M^2+\mu}{\mu}\|\beta_1 - \beta_0\|_2.$$

Hence, for arbitrary $\varepsilon > 0$, if

$$\left(\frac{\|M^2\mathbf{I} - \mathbf{H}^{\intercal}\mathbf{H}\|}{M^2+\mu}\right)^k \frac{M^2+\mu}{\mu}\|\beta_1 - \beta_0\|_2 < \varepsilon,$$

that is,

$$k > \frac{\log\left(\frac{\|\beta_1-\beta_0\|_2(M^2+\mu)}{\varepsilon\mu}\right)}{\log\left(\frac{M^2+\mu}{\|M^2\mathbf{I}-\mathbf{H}^{\intercal}\mathbf{H}\|}\right)},$$

then

$$\|\beta^{\star} - \beta_k\|_2 < \varepsilon. \tag{2.6}$$

3 Algorithm and Analysis

Based on Theorem 2.1 and Remark 2.1, we propose an iterative learning algorithm for training FNNRWs, denoted by IFNNRW for simplicity. In our proposed IFNNRW, we suppose that activation functions take values of $(0, 1)$.

3.1 Algorithm description

Algorithm 1. *Given a training set $\mathscr{T} = \{(\mathbf{x}_i, t_i) : i = 1, 2, \ldots, N\}$, activation function g, hidden node number L, regularizing factor $\mu > 0$, and an acceptable accuracy ε.*

***Step** 1:* *Randomly assign input weights ω_i and biases b_i $(i = 1, 2, \ldots, L)$ in a proper scope.*

***Step** 2:* *Calculate the hidden layer output matrix $\mathbf{H}$.*

***Step** 3:* *Let $\beta_0 = (1, \ldots, 1)^\intercal$, $\beta_1 = \frac{1}{(NL^2+\mu)}((NL^2\mathbf{I} - \mathbf{H}^\intercal\mathbf{H})\beta_0 + \mathbf{H}^\intercal\mathbf{T})$, $k := 1$ and K be a minimal positive integer larger than*

$$\frac{\log\left(\frac{\|\beta_1-\beta_0\|_2(NL^2+\mu)}{\varepsilon\mu}\right)}{\log\left(\frac{NL^2+\mu}{\|NL^2\mathbf{I}-\mathbf{H}^\intercal\mathbf{H}\|}\right)}.$$

***Step** 4:* *If $k \geq K$, stop; else $k := k+1$ and update β as follows*

$$\beta_{k+1} = \frac{1}{NL^2+\mu}\left((NL^2\mathbf{I} - \mathbf{H}^\intercal\mathbf{H})\beta_k + \mathbf{H}^\intercal\mathbf{T}\right)$$

go onto Step 4. The required output weight is $\beta_\varepsilon := \beta_K$.

3.2 Convergence analysis

Theorem 3.1. *Let β_ε be given in Algorithm 1. Then β_ε strongly converges to the minimizer of the following minimization problem as $\varepsilon \to 0$:*

$$\min_{\beta\in\mathbf{R}^L}\left\{\|\mathbf{H}\beta - \mathbf{T}\|_2^2 + \mu\|\beta\|_2^2\right\} \tag{3.1}$$

Remark 3.1. *The proposed IFNNRW algorithm solves the regularization model of FNNs and combines FNNRW with the iterative method, which guarantees the convergence of the algorithm theoretically. Furthermore, it overcomes the shortcomings of the FNNRW algorithm for the regularization model which cannot be applied to all $\mu > 0$. In addition, it is noteworthy that the constraint of the activation function is almost removed and IFNNRW thus can be used for any kind of activation functions provided that its function values are bounded. From the proof of Theorem 2.1, we can also conclude the convergence rate of IFNNRW, as follows:*

$$\frac{\|\beta_{k+1} - \beta_k\|_2}{\|\beta_k - \beta_{k-1}\|_2} \leq \frac{\|NL^2\mathbf{I} - \mathbf{H}^\intercal\mathbf{H}\|}{NL^2+\mu}. \tag{3.2}$$

From (3.2), we observe that the larger N and L are, the slower the convergence rate will be. Therefore, we need to normalize the activation functions by multiplying a factor $1/(\sqrt{N}L)$. In this case, $\|\mathbf{H}\| \leq 1$ and the number K of iterations would thus be much smaller. In fact, from Remark 2.1 and $\|NL^2\mathbf{I} - \mathbf{H}^\intercal\mathbf{H}\| \leq 1$, we have

$$K > \frac{\log\left(\frac{\|\beta_1-\beta_0\|_2(NL^2+\mu)}{\varepsilon\mu}\right)}{\log\left(\frac{NL^2+\mu}{\|NL^2\mathbf{I}-\mathbf{H}^\intercal\mathbf{H}\|}\right)} \geq \frac{\log\left(\frac{\|\beta_1-\beta_0\|_2}{\varepsilon\mu}\right)}{\log(1+\mu)} + 1.$$

Hence, we modify Algorithm 1 as follows.

Algorithm 2. *Given a training set $\mathscr{T} = \{(\mathbf{x}_i, t_i) : i = 1, 2, \ldots, L\}$, an activation function g, a hidden node number L, a regularizing factor $\mu > 0$, and an acceptable accuracy ε.*

***Step** 1: Randomly assign input weights ω_i and biases b_i ($i = 1, 2, \ldots, L$) in a proper scope.*

***Step** 2: Calculate the hidden layer output matrix $\mathbf{H}$ and modify it by multiplying the factor $1/(\sqrt{N}L)$. That is, $\mathbf{H} \to 1/(\sqrt{N}L)\mathbf{H}$. (Hence, $\|\mathbf{H}\| \leq 1$)*

***Step** 3: Let $\beta_0 = (1, \ldots, 1)^\intercal$, $\beta_1 = \frac{1}{1+\mu}\left((\mathbf{I} - \mathbf{H}^\intercal\mathbf{H})\beta_0 + \mathbf{H}^\intercal\mathbf{T}\right)$, $k := 1$ and K be minimal positive integer larger than*

$$\frac{\log\left(\frac{\|\beta_1-\beta_0\|_2}{\varepsilon\mu}\right)}{\log(1+\mu)} + 1.$$

***Step** 4: If $k \geq K$, stop; else $k := k + 1$ and update β as follows*

$$\beta_{k+1} = \frac{1}{1+\mu}\left((\mathbf{I} - \mathbf{H}^\intercal\mathbf{H})\beta_k + \mathbf{H}^\intercal\mathbf{T}\right)$$

go onto Step 4. The required output weight is $\beta_\varepsilon := \beta_K$.

3.3 Stability analysis

It is interesting to see if our estimation of the output weights obtained by the IFNNRW problem converges to the ideal solution $\beta_0^\star$ when the image $\mathbf{H}\beta_0^\star$ has a sufficiently small perturbation from the original data $\mathbf{T}$. Since the minimizer $\beta_\delta^\star$ of

$$\phi_{\delta;\mathbf{T}}(\beta) := \|\mathbf{H}\beta - \mathbf{T}\|_2^2 + \delta\mu\|\beta\|_2^2$$

differs from $\beta_0^\star$ if $\delta \neq 0$, we should include the condition of $\delta \to 0$. We state the concept of stability of the estimate as follows, which was established by Daubechies et al. [10, Section 4]. For each $\beta_0^\star$ in a certain class of functions, there exists a function $\delta(e)$ of the noise level e, such that $\delta(e) \to 0$ and

$$\sup_{\|\mathbf{T}-\mathbf{H}\beta_0^\star\|_2 \leq e} \left\| \beta_{\delta(e);\mathbf{T}}^\star - \beta_0^\star \right\|_2 \to 0$$

as $e \to 0$, where $\beta_{\delta(e);\mathbf{T}}^\star$ is the minimizer of

$$\phi_{\delta(e);\mathbf{T}}(\beta) := \|\mathbf{H}\beta - \mathbf{T}\|_2^2 + \delta(e)\mu\|\beta\|_2^2.$$

Before proceeding, we establish the following lemma (its proof is given in the Appendix).

Lemma 3.1. *Let $L \geq 1$ and $\Phi_\mu(\beta)$, $\beta \in \mathbf{R}^L$, be given in* (2.5) *and $\{\beta_k\}_{k=0}^\infty$ be the sequence of $\mathbf{R}^L$ given by Theorem 2.1. Under the condition of Theorem 2.1, $\Phi_\mu(\beta_k)$ is a non-increasing sequence with respect to k.*

Remark 3.2. *Lemma 3.1 implies*

$$\mu\|\beta_k\|_2^2 \leq \Phi_\mu(\beta_k) \leq \Phi_\mu(\beta_0),$$

i.e., $\mu\|\beta_k\|_2^2$ is bounded.

The following theorem characterizes the stability of our IFNNRW algorithm. The framework of the proof, which is included in the Appendix, comes from [10].

Theorem 3.2. *Let L, N be positive integers and let $\mathbf{H}$ be a matrix in $\mathbf{R}^{N\times L}$ satisfying $\|\mathbf{H}\| < M$ for some positive constant M. Let $\delta(e) : (0, +\infty) \to (0, +\infty)$ be a function of e satisfying*

$$\lim_{e\to 0} \delta(e) = 0 \quad \textit{and} \quad \lim_{e\to 0} e^2/\delta(e) = 0.$$

Then for $\beta_0^\star \in \mathbf{R}^L$, we have

$$\lim_{e\to 0}\left(\sup_{\|\mathbf{T}-\mathbf{H}\beta_0^\star\|_2 \leq e} \|\beta_{\delta(e);\mathbf{T}}^\star - \beta^\dagger\|_2 \right) = 0, \tag{3.3}$$

where $\beta_{\delta(e);\mathbf{T}}^\star$ is a minimizer of $\|\mathbf{T} - \mathbf{H}\beta\|_2^2 + \delta(\epsilon)\mu\|\beta\|_2^2$ and $\beta^\dagger$ is the unique element in $\Gamma = \{\beta \in \mathbf{R}^L : \mathbf{H}\beta = \mathbf{H}\beta_0^\star\}$ minimizing the norm $\|\cdot\|_2$ and the supreme in (3.3) *takes over all $\mathbf{T} \in \mathbf{R}^N$ satisfying $\|\mathbf{T} - \mathbf{H}\beta_0^\star\|_2 \leq e$ for some $e > 0$.*

Remark 3.3. *The newly proposed algorithm requires few constraints on the activation functions and thus can train the FNNs with various kinds of activation functions. The only constraint on the activation function is that its function values lie in* $(0, 1)$. *As a matter of fact, this condition can be weakened so that the function is bounded. Namely, if the activation functions are absolutely bounded by a positive constant* C*, then via a small modification of the algorithm IFNNRW, that is, by taking*

$$\beta_{k+1} = \frac{1}{CNL^2 + \mu} \big((CNL^2\mathbf{I} - \mathbf{H}^\intercal\mathbf{H})\beta_k + \mathbf{H}^\intercal\mathbf{T}\big)$$

(replacing NL^2 *by* CNL^2*) in the fourth step of the algorithm, all the conclusions about the convergence and stability of IFNNRW hold as well.*

4 Performance Evaluation

In this section, the performance of the proposed IFNNRW learning algorithm is measured. The simulations for the IFNNRW algorithm are carried out in the MATLAB 7.10.0 (2010a) environment, running a Pentium(R) Dual-Core E5400 processor with a speed of 2.70 GHz. The activation function used in the new algorithm is the sigmoidal function $g(x) = 1/(1 + e^{-x})$. It has been observed from our experience that the range setting of random input weights and biases has a great impact on the system's performance. By referring to the formulation in [16], it is necessary to estimate the scope of the random input weights and biases. However, we simplified the proposed setting as an interval $[-\alpha, \alpha]$, where parameter α is data-dependent and can be determined empirically.

4.1 UCI datasets

In order to further check the IFNNRW algorithm, the effectiveness of the proposed new algorithm is verified by data in real life. The new algorithm is tested for three real classification problems, namely Diabetes, Glass Identification (Glass ID), and MAGIC Gamma Telescope, which are detailed in Table 1. All three datasets are from the UCI repository of machine learning databases [5]. For each database,we conduct fifty trials on each of the training and testing samples and then take the average as the final result to reduce the random error. The results of each database, compared with FNNRW and

Table 1: Information on the UCI datasets

Datasets	Training	Testing	Attributes	Associated Tasks
Diabetes	378	390	9	Classification
Glass ID	73	73	10	Classification
MAGIC Gamma Telescope	19020	9423	11	Classification

the Closed-form solution to the ℓ_2 regularization model, are shown in Table 2, Figure 1 and Figure 2, which indicates that the IFNNRW algorithm gains good results on real databases. Moreover, in order to observe the computational efficiency, a comparison of the required training time for the different learning algorithms on Diabetes and Glass ID is listed in Figure 3.

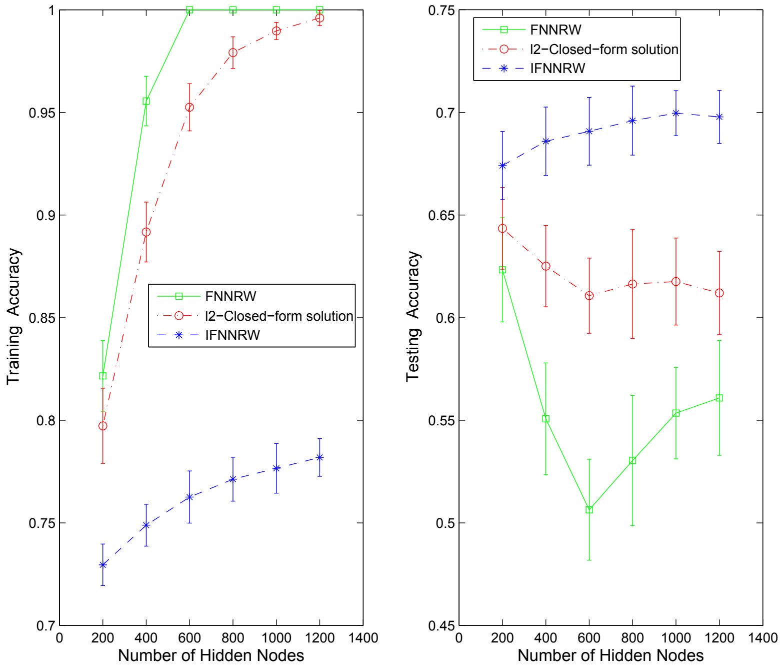

Fig. 1: The accuracy and the corresponding standard deviation of the three methods for Diabetes.

For the three databases, we set the regularizing factor $\mu = 0.5$, $\alpha = 2$ and the acceptable accuracy $\varepsilon = 0.001$. The accuracy and the corresponding standard deviation of training and testing curves on the first two UCI databases are given in Figure 1 and

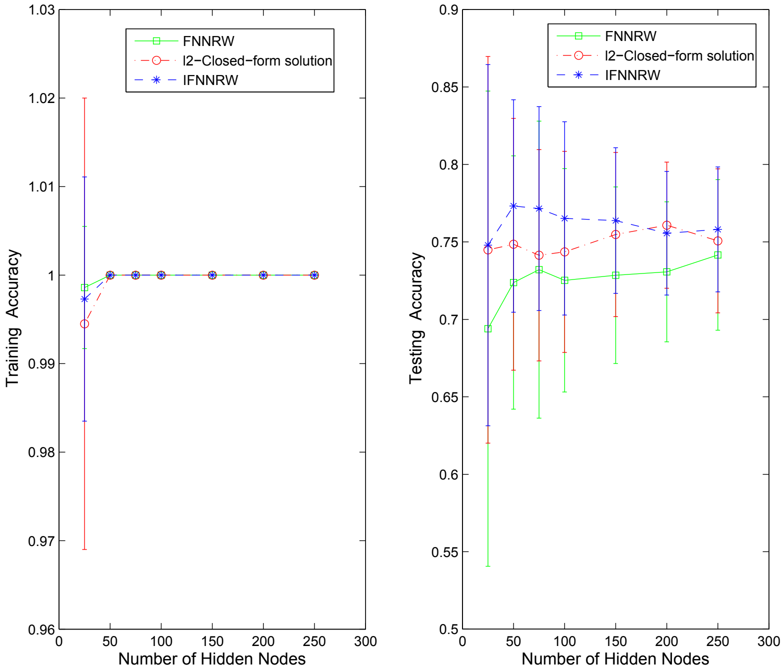

Fig. 2: The accuracy and the corresponding standard deviation of the three methods for Glass ID.

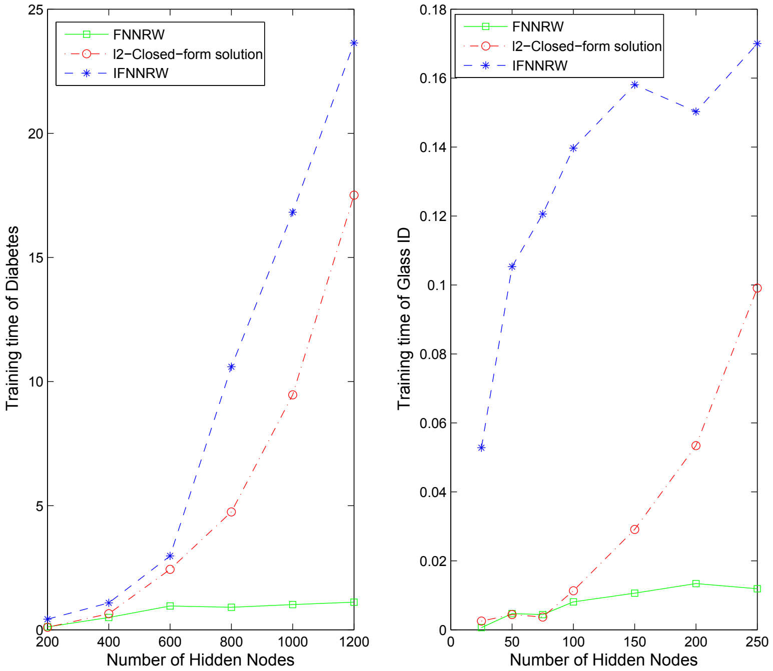

Fig. 3: The required training time on the three methods for Diabetes and Glass ID.

Table 2: The accuracy of MAGIC Gamma telescope

Nodes	FNNRW		ℓ_2-Closed-form solution		IFNNRW	
	Training	Testing	Training	Testing	Training	Testing
800	out of memory	out of memory	out of memory	out of memory	0.7593	0.7539
850	out of memory	out of memory	out of memory	out of memory	0.7584	0.7531
900	out of memory	out of memory	out of memory	out of memory	0.7600	0.7552
950	out of memory	out of memory	out of memory	out of memory	0.7592	0.7544
1000	out of memory	out of memory	out of memory	out of memory	0.7600	0.7546

Figure 2, respectively. Using the data in the figures, we compare the accuracy of the testing and standard deviation of FNNRW and the Closed-form solution to the ℓ_2 regularization model with the proposed IFNNRW algorithm and draw the following conclusions to the UCI classification problem:

(i) Comparing the accuracy of FNNRW, the ℓ_2-Closed-form solution and IFNNRW, it can be seen that the testing accuracy of IFNNRW is higher than the other two, which illustrates that the iterative ℓ_2 regularization method achieves better generalization performance. Also, the corresponding standard deviation of the testing is too small, indicating that the new algorithm is much more stable than the others. Also, when the number of neurons increases, FNNRW will experience the problem of over-fitting, that is to say, a high recognition rate in the training and a low rate on the testing. However, IFNNRW can effectively avoid the over-fitting problem. Another problem is that the solution to the proposed IFNNRW algorithm has been theoretically proved to converge to the ℓ_2-Closed-form solution in Theorem 2.1. Nevertheless, in our experiments, especially for the Diabetes datasets, it seems that IFNNRW and the ℓ_2-Closed-form solutions are converging to different solutions. A possible explanation for this is that the real dataset itself stemming from the UCI repository of machine learning databases contains some structural problems, which influence the iterative solution to the proposed algorithm to some extent.

(ii) Most of the time, using FNNRW to calculate the Moore-Penrose generalized inverse of the output matrix $\mathbf{H}_{N\times L}$ consumes a large amount of memory. Generally, if the number of samples N is large enough, then it should increase the number of hidden neurons L in order to obtain a higher training rate, which leads to the problem of calculating a large scale Moore-Penrose generalized inverse of $\mathbf{H}_{N\times L}$ for FNNRW. Table 2 indicates that INNRW does not perform well when the number of hidden neurons increases to 800, and the MATLAB displays runs out of memory. This is due to the fact that the Moore-Penrose generalized inverse of the scale output matrix $\mathbf{H}_{N\times L}$ cannot be calculated in this computational environment. Fortunately, this is not a problem for IFNNRW. It can continue to calculate when the number of hidden neurons is increasing, and the results are stable. Thus, IFNNRW is more effective than FNNRW for computing large scale data, and has satisfactory accuracy. Further, it can be observed from Figure 3 that the computational time of IFNNRW on training is higher than FNNRW and its ℓ_2-Closed-form solution methods. The reason for this is that the iterative IFNNRW method is more complex in its computing process compared with the other two methods which use a matrix operation. Overall, the proposed IFNNRW algorithm is successful and stable for classification.

4.2 Face recognition dataset

In this subsection, we evaluate the performance of the proposed IFNNRW in a face recognition experiment preprocessed by fast discrete curvelet transform [6] and 2-dimensional principle component analysis (2DPCA) [32], both of which are usually efficient methods for image feature extraction [12, 15].

Taking the YALE database [2] which contains images from fifteen individuals, each providing eleven different images as an example, we divide these 165 pictures into two sets: the training set and the testing set. Then we use fast discrete curvelet transform and 2DPCA to extract the main projection features of these images, and finally utilize the IFNNRW algorithm to classify the features to complete the face recognition. Here, the pixels in each image are vectorized into a d-dimensional vector $\mathbf{x}_i$, and the vectors obtained in this manner from all ℓ-classes divided by n sample images will be denoted as $\mathbf{X} = \{\mathbf{x}_1^1, \mathbf{x}_2^1, \ldots, \mathbf{x}_{n_\ell}^\ell\}$.

Through experimentation, we extract five main projection features and specify the regularizing factor $\mu = 0.5$ and the scope parameter $\alpha = 5$ to carry out the experiment.

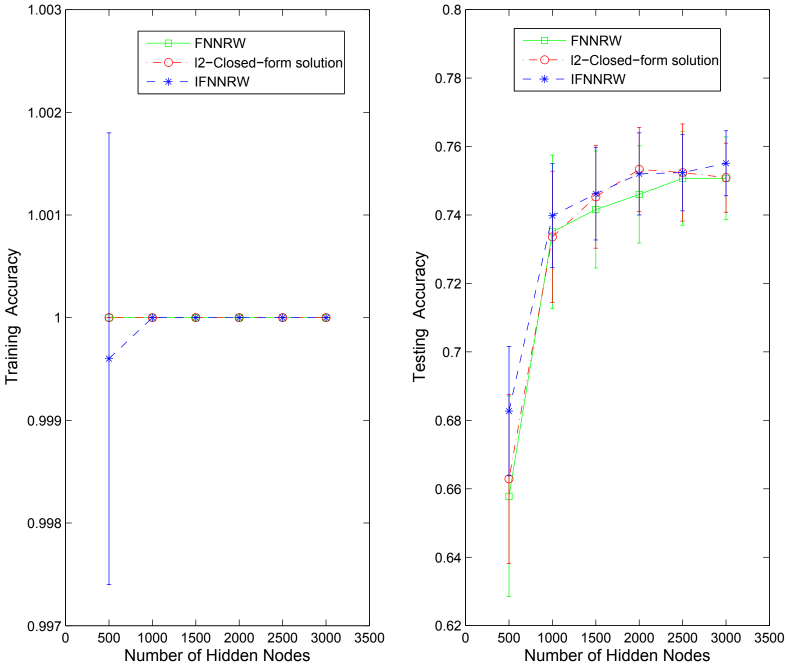

Fig. 4: The accuracy and the corresponding standard deviation of the three methods for Yale.

In order to reduce the random error, we conduct fifty trials, and then take the average as the final result. Moreover, we set the maximum iteration to 1000 to speed up the convergence. The result of the accuracy and standard deviation for the three methods is shown in Figure 4, which indicates that in our simulation, on average, the INNRW algorithm has both satisfactory recognition rate and stability.

Comparing the accuracy of the testing and the standard deviation of FNNRW, the Closed-form solution to the regularization model, and the IFNNRW algorithm, we can see that the IFNNRW algorithm has a good generalization ability and has more stable performance as shown in Figure 4, which is similar to the analysis of the UCI databases. Thus, the reason why the proposed IFNNRW algorithm is effective in the experiments on the face recognition problem lies in the regularization model and the iterative method which can, as shown in Section 3.3, stably solve the linear inverse problem.

5 Concluding Remarks

This paper proposed a randomized learning algorithm based on the iterative method for training FNNs. The proposed algorithm demonstrates some interesting features, including:

- IFNNRW maintains the first two steps of FNNRW, i.e., the random assignment of input weights and biases and the calculation of the hidden layer output matrix, but in the third step it uses the iterative method to calculate the output weights which can approximate the minimizer of the regularization model

$$\|\mathbf{H}\beta - \mathbf{T}\|_2^2 + \mu\|\beta\|_2^2.$$

- In this paper, we mathematically proved the convergence and the stability of the IFNNRW algorithm, and obtained the upper bound of the rate of convergence, which in theory, ensures that IFNNRW can train the FNNs in the given circumstances effectively.

Admittedly, further efforts should be made so that the proposed algorithm can overcome its own shortcomings. For instance, the IFNNRW algorithm cannot approximate the regression samples very well in our experiments, which are not presented in the paper. A logical explanation for this phenomenon lies in the fact that the regression problem does not satisfy the stability condition and is likely to make the algorithm divergent. How to explain this point and improve the performance of the IFNNRW algorithm still remains open. Also, it should be noted that in our experiments, the choice of the regularizing factor μ and acceptable error setting have a considerable impact on the results, and often a small μ can help to achieve the desired results. Thus, it is interesting to know how to properly select these parameters so that the performance of IFNNRW can be improved.

Another valuable point we should be aware is that there are various choices of the additive item $J(\beta)$ in (2.3) which can be selected as $J(\beta) = \sum_{\gamma=1}^{\infty} w_\gamma |\langle \beta, \varphi_\gamma \rangle|^p$ where w_γ is a strictly positive sequence and $1 \leq p \leq 2$ (see [10]). For instance, if $p = 1$ and $w_\gamma = \mu$, then (2.3) actually becomes the ℓ_1 regularization model which is a significant and meaningful model of compressive sensing and promotes the sparsity of the problem.

6 Appendix

The proofs of Theorem 2.1 and Lemma 3.1, using a surrogate functional (see $\Phi_\mu^{\mathrm{SUR}}(\beta,\gamma)$ below), use the argument from [10].

Proof of Theorem 2.1. Since $\|\mathbf{H}\| < M$, $\|M^2\mathbf{I} - \mathbf{H}^\intercal\mathbf{H}\| \le M^2$, we have

$$\begin{aligned}\|\beta_{k+1} - \beta_k\|_2 &= \frac{1}{M^2+\mu}\big\|(M^2\mathbf{I} - \mathbf{H}^\intercal\mathbf{H})(\beta_k - \beta_{k-1})\big\|_2 \\ &\le \frac{M^2}{M^2+\mu}\|\beta_k - \beta_{k-1}\|_2 \le \left(\frac{M^2}{M^2+\mu}\right)^k \|\beta_1 - \beta_0\|_2,\end{aligned}$$

and, for arbitrary positive integer q,

$$\begin{aligned}&\|\beta_{k+q} - \beta_k\|_2 \le \|\beta_{k+q} - \beta_{k+q-1}\|_2 + \|\beta_{k+q-1} - \beta_{k+q-1}\|_2 + \cdots + \|\beta_{k+1} - \beta_k\|_2 \\ &\le \left(\left(\frac{M^2}{M^2+\mu}\right)^{k+q-1} + \left(\frac{M^2}{M^2+\mu}\right)^{k+q-2} + \cdots + \left(\frac{M^2}{M^2+\mu}\right)^{k}\right)\|\beta_1 - \beta_0\|_2 \\ &\le \left(\frac{M^2}{M^2+\mu}\right)^k \frac{M^2+\mu}{\mu}\|\beta_1 - \beta_0\|_2 \to 0, \quad k \to \infty.\end{aligned}$$

Therefore, $\{\beta_k\}_{k=1}^\infty$ is a Cauchy sequence. Since $\mathbf{R}^L$ is a complete Banach space, $\{\beta_k\}_{k=1}^\infty$ is strongly convergent.

Let

$$\begin{aligned}\Phi_\mu(\beta) &:= \|\mathbf{H}\beta - \mathbf{T}\|_2^2 + \mu\|\beta\|_2^2, \\ \Phi_\mu^{\mathrm{SUR}}(\beta,\gamma) &:= \Phi_\mu(\beta) + M^2\|\beta - \gamma\|_2^2 - \|\mathbf{H}\beta - \mathbf{H}\gamma\|_2^2, \end{aligned} \tag{6.1}$$

and

$$\mathbf{\Psi}\beta := \frac{1}{M^2+\mu}\left((M^2\mathbf{I} - \mathbf{H}^\intercal\mathbf{H})\beta + \mathbf{H}^\intercal\mathbf{T}\right). \tag{6.2}$$

Since $\mathbf{H}$ and $\mathbf{H}^\intercal$ are bounded linear operators, $\{\beta_k\}$ strongly converges to a fixed point of $\mathbf{\Psi}$, that is, there exists $\beta^\star$ such that $\|\beta_k - \beta^\star\|_2 \to 0$ as $k \to \infty$ and $\mathbf{\Psi}\beta^\star = \beta^\star$.

Now, we prove that $\beta^\star$ is the minimizer for $\Phi_\mu(\beta)$. By (6.1),

$$\begin{aligned}\Phi_\mu^{\mathrm{SUR}}(\beta,\beta^\star) &= \Phi_\mu(\beta) + M^2\|\beta - \beta^\star\|_2^2 - \|\mathbf{H}\beta - \mathbf{H}\beta^\star\|_2^2 \\ &= (M^2+\mu)\|\beta\|_2^2 - 2\langle\beta, M^2\beta^\star + \mathbf{H}^\intercal(\mathbf{T} - \mathbf{H}\beta^\star)\rangle \\ &\quad + \|\mathbf{T}\|_2^2 + M^2\|\beta^\star\|_2^2 - \|\mathbf{H}\beta^\star\|_2^2.\end{aligned} \tag{6.3}$$

This implies that

$$\frac{1}{(M^2+\mu)}(M^2\beta^\star + \mathbf{H}^\intercal(\mathbf{T} - \mathbf{H}\beta^\star)) = \mathbf{\Psi}\beta^\star = \beta^\star$$

is a minimizer of $\Phi_\mu^{\mathrm{SUR}}(\beta,\beta^\star)$, where the first equality uses (6.2). This with (6.3) then gives

$$\Phi_\mu^{\mathrm{SUR}}(\beta,\beta^\star) = (M^2+\mu)\|\beta-\beta^\star\|_2^2 + \|\mathbf{T}\|_2^2 - \mu\|\beta^\star\|_2^2 - \|\mathbf{H}\beta^\star\|_2^2.$$

Then, for any $h \in \mathbf{R}^L$,

$$\Phi_\mu^{\mathrm{SUR}}(\beta^\star+h,\beta^\star) \geq \Phi_\mu^{\mathrm{SUR}}(\beta^\star,\beta^\star) + M^2\|h\|_2^2 = \Phi_\mu(\beta^\star) + M^2\|h\|_2^2, \tag{6.4}$$

where the equality uses (6.1). Using (6.1) again,

$$\Phi_\mu^{\mathrm{SUR}}(\beta^\star+h,\beta^\star) = \Phi_\mu(\beta^\star+h) + M^2\|h\|_2^2 - \|\mathbf{H}h\|_2^2,$$

which with (6.4) gives

$$\Phi_\mu(\beta^\star+h) \geq \Phi_\mu(\beta^\star) + \|\mathbf{H}h\|_2^2 = \Phi_\mu(\beta^\star),$$

completing the proof. □

Proof of Lemma 3.1. From the proof of Theorem 2.1, $\big(M^2\beta_k+\mathbf{H}^\intercal(\mathbf{T}-\mathbf{H}\beta_k)\big)/(M^2+\mu) = \beta_{k+1}$ is a minimizer of

$$\begin{aligned}\Phi_\mu^{\mathrm{SUR}}(\beta,\beta_k) = (M^2+\mu)\|\beta\|_2^2 - 2\langle\beta, M^2\beta_k + \mathbf{H}^\intercal(\mathbf{T}-\mathbf{H}\beta_k)\rangle \\ + \|\mathbf{T}\|_2^2 + M^2\|\beta_k\|_2^2 - \|\mathbf{H}\beta_k\|_2^2.\end{aligned}$$

By $\|\mathbf{H}\| < M$ and (6.1),

$$\begin{aligned}\Phi_\mu(\beta_{k+1}) &\leq \Phi_\mu(\beta_{k+1}) + M^2\|\beta_k-\beta_{k+1}\|_2^2 - \|\mathbf{H}(\beta_k-\beta_{k+1})\|_2^2 \\ &= \Phi_\mu^{\mathrm{SUR}}(\beta_{k+1},\beta_k) \leq \Phi_\mu^{\mathrm{SUR}}(\beta_k,\beta_k) = \Phi_\mu(\beta_k),\end{aligned}$$

thus completing the proof. □

The proof of Theorem 3.2. The proof uses the argument of [10, Section 4]. For $\delta > 0$, let

$$\phi_{\delta;\mathbf{T}}(\beta) := \|\mathbf{H}\beta-\mathbf{T}\|_2^2 + \delta\mu\|\beta\|_2^2. \tag{6.5}$$

Since $\phi_{\delta;\mathbf{T}}(\beta)$ is strictly convex with respect to β, it has a unique minimizer which we denote by $\beta_{\delta;\mathbf{T}}^\star$.

We first establish the weak convergence. Let us prove that if $(\mathbf{T}_l)_{l=1}^\infty$ is a sequence of $\mathbf{R}^L$ such that $\|\mathbf{T}_l - \mathbf{H}\beta_0^\star\|_2 \leq e_l \to 0$, as $l\to\infty$, then $\beta_{\delta(e_l);\mathbf{T}_l}^\star$ weakly converges to

$\beta^\dagger$. In the proof, we denote $\delta_{e_l} := \delta(e_l)$ for simplicity. By (6.5), let $\beta^\star$ be a minimizer of (6.5) with $\delta := \delta_{e_l}$ and $\mathbf{T} := \mathbf{T}_l$,

$$\begin{aligned}\left\|\beta^\star_{\delta_{e_l};\mathbf{T}_l}\right\|_2^2 &\le \frac{1}{\mu\delta_{e_l}}\phi_{\delta_{e_l};\mathbf{T}_l}\left(\beta^\star_{\delta_{e_l};\mathbf{T}_l}\right)\\ &\le \frac{1}{\mu\delta_{e_l}}\phi_{\delta_{e_l};\mathbf{T}_l}\left(\beta^\dagger\right)\\ &= \frac{1}{\mu\delta_{e_l}}\left(\|\mathbf{H}\beta_0^\star - \mathbf{T}_l\|_2^2 + \delta_{e_l}\|\beta^\dagger\|_2^2\right)\\ &\le \frac{1}{\mu}\left(\frac{e_l^2}{\delta_{e_l}} + \|\beta^\dagger\|_2^2\right). \end{aligned} \tag{6.6}$$

This shows that $\left\|\beta^\star_{\delta_{e_l};\mathbf{T}_l}\right\|_2^2$ is bounded. Hence, there exists at least one weak accumulation limit of $\beta^\star_{\delta_{e_l};\mathbf{T}_l}$, and there thus exists a subsequence $\left(\beta^\star_{\delta_{e_{l_{c(k)}}};\mathbf{T}_{l_{c(k)}}}\right)_{k=1}^\infty$ of $(\beta^\star_{\delta_{e_l};\mathbf{T}_l})_{l=1}^\infty$ that has a weak limit $\widetilde{\beta}$. For simplicity, we write this subsequence as

$$(\widetilde{\beta}_k)_{k=1}^\infty = \left(\beta^\star_{\delta_{e_{l_{c(k)}}};\mathbf{T}_{l_{c(k)}}}\right)_{k=1}^\infty,$$

and

$$\widetilde{\mathbf{T}}_k = \mathbf{T}_{l_{c(k)}}, \quad \widetilde{\delta}_k = \delta_{c(k)}, \quad \widetilde{e}_k = \widetilde{\mathbf{T}}_k - \mathbf{H}\beta_0, \quad \widetilde{e}_k = e_{l_{c(k)}},$$

which satisfy

$$\widetilde{\delta}_k \to 0,\ \left(\widetilde{e}_k\right)^2/\widetilde{\delta}_k \to 0 \ \text{ as } k\to\infty \quad \text{ and } \quad \|\widetilde{e}_k\|_2 \le \widetilde{e}_k,\ \ k \ge 1. \tag{6.7}$$

Now, we prove that $\widetilde{\beta} = \beta^\dagger$. Let

$$F_k(\beta) := \frac{M^2\beta + \mathbf{H}^\intercal(\widetilde{\mathbf{T}}_k - \mathbf{H}\beta)}{M^2 + \widetilde{\delta}_k\mu}$$

be an operator from $\mathbf{R}^L$ to $\mathbf{R}^N$. By the proof of Theorem 2.1, any fixed point of operator F_k is a minimizer of $\phi_{\widetilde{\delta}_k;\widetilde{\mathbf{T}}_k}(\beta)$. Since $\widetilde{\beta}_k$ is the unique minimizer of $\phi_{\widetilde{\delta}_k;\widetilde{\mathbf{T}}_k}(\beta)$, then $\widetilde{\beta}_k$ is a fixed point of F_k.

Let $(\varphi_\gamma)_{\gamma=1}^L$ be an orthonormal basis of $\mathbf{R}^L$ and let $\beta_\gamma := \langle\beta,\varphi_\gamma\rangle$ be the Fourier coefficients of a vector β in $\mathbf{R}^L$. Then

$$\begin{aligned}\left(\widetilde{\beta}\right)_\gamma &= \lim_{k\to\infty}\langle\widetilde{\beta}_k,\varphi_\gamma\rangle = \lim_{k\to\infty}\langle F_k(\widetilde{\beta}_k),\varphi_\gamma\rangle\\ &= \lim_{k\to\infty}\frac{\left\langle M^2\widetilde{\beta}_k + \mathbf{H}^\intercal(\widetilde{\mathbf{T}}_k - \mathbf{H}\widetilde{\beta}_k),\varphi_\gamma\right\rangle}{M^2 + \widetilde{\delta}_k\mu}\\ &=: \lim_{k\to\infty}\frac{(h_k)_\gamma}{M^2 + \widetilde{\delta}_k\mu}, \end{aligned} \tag{6.8}$$

where we let

$$\widetilde{h}_k := M^2\widetilde{\beta}_k + \mathbf{H}^\intercal(\widetilde{\mathbf{T}}_k - \mathbf{H}\widetilde{\beta}_k) = M^2\widetilde{\beta}_k + \mathbf{H}^\intercal\mathbf{H}(\beta_0^\star - \widetilde{\beta}_k) + \mathbf{H}^\intercal\widetilde{e}_k. \tag{6.9}$$

By Remark 3.2, given a penalty weight μ, norms $\|\widetilde{\beta}_k\|_2$ and $\|\beta_0^\star\|_2$ are bounded by a positive constant $C := C_\mu$. This with $\|\mathbf{H}\| < M$ and Cauchy-Schwartz inequality gives

$$\begin{aligned}\left|(\widetilde{h}_k)_\gamma\right| &= \left|\left\langle M^2\widetilde{\beta}_k + \mathbf{H}^\intercal\mathbf{H}(\beta_0^\star - \widetilde{\beta}_k) + \mathbf{H}^\intercal\widetilde{e}_k, \varphi_\gamma\right\rangle\right| \\ &\leq M^2\|\widetilde{\beta}_k\|_2\|\varphi_\gamma\|_2 + \|\mathbf{H}^\intercal\|\|\mathbf{H}\|\|\beta_0^\star - \widetilde{\beta}_k\|_2 + \widetilde{e}_k\|\mathbf{H}^\intercal\|\|\varphi_\gamma\|_2 \\ &\leq 3M^2C + M\,\widetilde{e}_k,\end{aligned}$$

where we use (6.7) and $\|\varphi_\gamma\|_2 = 1$. Therefore, $(\widetilde{h}_k)_\gamma$ are bounded. Using (6.8) and $\widetilde{\delta}_k \to 0$ as $k \to \infty$ gives

$$M^2(\widetilde{\beta})_\gamma = \lim_{k\to\infty}\left(\frac{M^2}{M^2 + \widetilde{\delta}_k\mu} - 1\right)(\widetilde{h}_k)_\gamma + \lim_{k\to\infty}(\widetilde{h}_k)_\gamma = \lim_{k\to\infty}(\widetilde{h}_k)_\gamma.$$

This together with (6.9) and the weak convergence of $\widetilde{\beta}_k$ to $\widetilde{\beta}$ and the weak convergence of $\widetilde{e}_k$ to the zero vector gives

$$\begin{aligned}M^2(\widetilde{\beta})_\gamma &= \lim_{k\to\infty}(\widetilde{h}_k)_\gamma = \lim_{k\to\infty}\left(M^2(\widetilde{\beta}_k)_\gamma + \left\langle\mathbf{H}^\intercal\mathbf{H}\left(\beta_0^\star - \widetilde{\beta}_k\right), \varphi_\gamma\right\rangle + \langle\widetilde{e}_k, \mathbf{H}\varphi_\gamma\rangle\right) \\ &= M^2(\widetilde{\beta})_\gamma + \left(\mathbf{H}^\intercal\mathbf{H}\left(\beta_0^\star - \widetilde{\beta}\right)\right)_\gamma,\end{aligned}$$

for $\gamma = 1, \ldots, L$. Thus, $\mathbf{H}^\intercal\mathbf{H}\left(\beta_0^\star - \widetilde{\beta}\right) = 0$, which gives

$$\left\langle\mathbf{H}\left(\beta_0^\star - \widetilde{\beta}\right), \mathbf{H}\left(\beta_0^\star - \widetilde{\beta}\right)\right\rangle = \left\langle\mathbf{H}^\intercal\mathbf{H}\left(\beta_0^\star - \widetilde{\beta}\right), \beta_0^\star - \widetilde{\beta}\right\rangle = 0.$$

Then $\mathbf{H}\widetilde{\beta} = \mathbf{H}\beta_0^\star$. This implies $\widetilde{\beta} \in \Gamma := \{\beta \in \mathbf{R}^L : \mathbf{H}\beta = \mathbf{H}\beta_0^\star\}$. Since $\beta^\dagger$ has the smallest $\|\cdot\|_2$ norm among all β in Γ, $\|\widetilde{\beta}\|_2 \geq \|\beta^\dagger\|_2$.

On the other hand, since for all $\gamma = 1, \ldots, L$, $(\widetilde{\beta}_k)_\gamma \to (\widetilde{\beta})_\gamma$ as $k \to \infty$, the Fatou's Lemma with (6.6) then gives

$$\begin{aligned}\|\widetilde{\beta}\|_2^2 &= \sum_{\gamma=1}^{L}|\widetilde{\beta}_\gamma|^2 \leq \limsup_{k\to\infty}\sum_{\gamma=1}^{L}|(\widetilde{\beta}_k)_\gamma|^2 = \lim_{k\to\infty}\|\widetilde{\beta}_k\|_2^2 \\ &\leq \lim_{k\to\infty}\left(\frac{\widetilde{e}_k^2}{\widetilde{\delta}_k} + \|\beta^\dagger\|_2^2\right) = \|\beta^\dagger\|_2^2 \leq \|\widetilde{\beta}\|_2^2,\end{aligned} \tag{6.10}$$

where we use (6.7). Since $\beta^\dagger$ is the unique element in Γ minimizing the norm $\|\cdot\|_2$, $\widetilde{\beta} = \beta^\dagger$, we can similarly prove that any other weakly convergent subsequence of $\beta^\star_{\delta_{e_l};\mathbf{T}_l}$

has a weak limit. Thus, the sequence $\beta^{\star}_{\delta_{e_l};\mathbf{T}_l}$ itself weakly converges to $\beta^{\dagger}$ and similar to (6.10) we can then prove $\lim_{l\to\infty}\|\beta^{\star}_{\delta_{e_l};\mathbf{T}_l}\|_2 = \|\beta^{\dagger}\|_2$. This gives

$$\begin{aligned}\lim_{l\to\infty}\|\beta^{\star}_{\delta_{e_l};\mathbf{T}_l} - \beta^{\dagger}\|_2^2 &= \lim_{l\to\infty}\left(\|\beta^{\star}_{\delta_{e_l};\mathbf{T}_l}\|_2^2 + \|\beta^{\dagger}\|_2^2 - 2\left\langle\beta^{\star}_{\delta_{e_l};\mathbf{T}_l}, \beta^{\dagger}\right\rangle\right)\\ &= \|\beta^{\dagger}\|_2^2 + \|\beta^{\dagger}\|_2^2 - 2\left\langle\beta^{\dagger}, \beta^{\dagger}\right\rangle = 0.\end{aligned}$$

This completes the proof of Theorem 3.2. □

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Nos. 91330118 and 61272023) and The State Key Laboratory of Synthetical Automation for Process Industries (Northeastern University).

References

[1] M. Alhamdoosh, D. Wang, Fast decorrelated neural network ensembles with random weights, Information Sciences 264 (2014) 104-117.

[2] Available: cvc.yale.edu/projects/yalefaces/yalefaces.html. Yale Face DB: Yale University, 2007.

[3] 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 Trans. Infor. Theory 44(2) (1998) 525-536.

[4] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15(6) (2003) 1373-1396.

[5] C. Blake, C. Merz, UCI repository of machine learning databases, *http://www.ics.uci.edu/ ˜mlearn/MLRepository.html*, Department of Information and Computer Sciences, University of California, Irvine, USA, 1998.

[6] E. Candès , L. Demanet, D. Donoho, L. Ying, Fast discrete curvelet transforms, Multiscale Modeling and Simulation 5(3) (2006) 861-899.

[7] F. L. Cao, Y. P. Tan, M. M. Cai, Sparse algorithms of random weight networks and applications, Expert Sys. Appl. 41 (2014) 2457-2462.

[8] F. L. Cao, H. L. Ye, D. Wang, A probabilistic learning algorithm for robust modeling using neural networks with random weights, Information Sciences 313 (2015) 62-78.

[9] D. Comminiello, M. Scarpiniti, S. Scardapane, R. Parisi, A. Uncini, Improving nonlinear modeling capabilities of functional link adaptive filters, Neural Networks 69 (2015) 51-59.

[10] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Commun. Pure and Applied Math. 57(11) (2004) 1413-1457.

[11] S. Dehuri, S. B. Cho, A comprehensive survey on functional link neural networks and an adaptive PSO-BP learning for CFLNN, Neural Computing and Applications 19(2) (2010) 187-205.

[12] D. L. Donoho, M. R. Duncan, Digital curvelet transform: Strategy, implementation, and experiments, Proc. of SPIE 4056 (2000) 12-30.

[13] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1990.

[14] A. N. Gorban, I. Y. Tyukin, D. V. Prokhorov, and K. I. Sofeikov, Approximation with random bases: Pro et contra, arXiv preprint arXiv:1506.04631, 2015.

[15] H. F Hu, Variable lighting face recognition using discrete wavelet transform, Pattern Recognition Letters 32 (2011) 1526-1534.

[16] B. Igelnik, Y. H. Pao, Stochastic choice of basis functions in adaptive function approximation and the functional-link net, IEEE Trans. Neural Networks 6(6) (1995) 1320-1329.

[17] T. T. Lee, J. T. Jeng, The chebyshev-polynomials-based unified model neural networks for function approximation, IEEE Trans. Sys., Man, and Cyber., Part B: Cybernetics 28(6) (1998) 925-935.

[18] A. M. Martinez, A. C. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intel. 23(2) (2011) 228-233.

[19] H. Moon, P. J. Phillips, Analysis of PCA-based face recognition algorithms, *Empirical Evaluation Techniques in Computer Vision*, by K.J. Bowyer and P.J. Phillips, eds., IEEE CS, 1998.

[20] Y. H. Pao, Y. Takefuji, Functional-link net computing: theory, system architecture, and functionalities, IEEE Computer, 25(5) (1992) 76-79.

[21] Y. H. Pao, G. H. Park, and D. J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6(2) (1994) 163-180.

[22] J. C. Patra, R. N. Pal, B. Chatterji, G. Panda, Identification of nonlinear dynamic systems using functional link artificial neural networks, IEEE Trans. Sys., Man, and Cyber., Part B: Cybernetics 29(2) (1999) 254-262.

[23] A. Pentland, T. Starner, N. Etcoff, et al., Experiments with eigenfaces, Proc. of Looking at People Workshop, Int'l Joint Conf. Artifical Intelligence, 1993.

[24] A. Rahimi, B. Recht, Uniform approximation of functions with random bases, Proc. of 46th Annual Allerton Conference on Communication, Control, and Computing. IEEE (2008) 555-561.

[25] A. Rahimi, B. Recht, Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning, Advances in Neural Information Processing Systems (2009) 1313-1320.

[26] C. Rao, S. K. Mitra, Generalized Inverse of Matrices and Its Applications, Wiley, New York, 1971.

[27] S. Scardapane, D. Wang, M. Panella, A. Uncini, Distributed learning for random vector functional-link networks, Information Sciences 301(20) (2015) 271-284.

[28] W. F. Schmidt, M. A. Kraaijveld, and R. P. W. Duin, Feed forward neural networks with random weights, Proc. of 11th IAPR Inter. Conf. Pattern Recog. Method. Sys. (1992) 1-4.

[29] C. Selina, S. Narayanan, K. C. C. Jay, Environmental sound recognition with time-frequency audio features, IEEE Trans. Audio, Speech, and Language Processing 17(6) (2009) 1142-1158.

[30] G. Wichern, J. C. Xue, H. Thornburg, Segmentation, indexing, and retrieval for environmental and natural sounds, IEEE Trans. Audio, Speech, and Language Processing 18(3) (2010) 688-707.

[31] H. K. Xu, Another control condition in an iterative method for nonexpansive mappings, Bull. Austral. Math. Soc. 65(1) (2002) 109-113.

[32] J. Yang, D. Zhang, A. F. Frangi, et al., Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE Trans. Pattern Anal. Mach. Intel. 26(1) (2004) 131-137.