IJCNN 2019. International Joint Conference on Neural Networks. Budapest, Hungary. 14-19 July 2019

An Initial Study on the Relationship Between Meta Features of Dataset and the Initialization of NNRW

Weipeng Cao
College of Computer Science and Software Engineering
Shenzhen University
Shenzhen, China
caoweipeng123@gmail.com

Muhammed J. A. Patwary
College of Computer Science and Software Engineering
Shenzhen University
Shenzhen, China
jamshed@szu.edu.cn

Pengfei Yang
State Key Laboratory of Computer Science, Institute of Software, CAS
University of Chinese Academy of Sciences
Beijing, China
yangpf@ios.ac.cn

Xizhao Wang
College of Computer Science and Software Engineering
Shenzhen University
Shenzhen, China
xzwang@szu.edu.cn

Zhong Ming*
College of Computer Science and Software Engineering
Shenzhen University
Shenzhen, China
mingz@szu.edu.cn

***Abstract*—The initialization of neural networks with random weights (NNRW) has a significant impact on model performance. However, there is no suitable way to solve this problem so far. In this paper, the relationship between meta features of a dataset and the initialization of NNRW is studied. Specifically, we construct seven regression datasets with known attributes' distributions, then initialize NNRW with different distributions and trained them based on the datasets to get the corresponding models respectively. The relationship between the attributes' distributions of the datasets and the initialization of NNRW is analyzed by the performance of the models. Several interesting phenomena are observed: firstly, initializing NNRW with the *Gaussian* distribution can help the model to have a faster convergence rate than ones with the *Gamma* and *Uniform* distribution. Secondly, if one or more attributes in a dataset that follow the *Gamma* distribution, using *Gamma* distribution to initialize NNRW may result in a slower convergence rate and easy overfitting. Thirdly, initializing NNRW with a specific distribution with smaller variances can always achieve faster convergence rate and better generalization performance than the one with larger variances. The above experimental results are not sensitive to the activation function and the type of NNRW. Some theoretical analyses about the above observations are also given in the study.**

Keywords*—Neural networks with random weights, extreme learning machine, random vector functional link network, meta feature

I. Introduction

NNRW (Neural Networks with Random Weights) is one of the special types of feedforward neural network, which are different from the traditional back propagation (BP) based neural networks in training mechanism [1]. In BP-based neural networks, all weights need to be iteratively fine-tuned until the model meets the expected accuracy. However, in NNRW, not all weights need to be trained. For example, for an NNRW with a single hidden layer, the input weights (i.e., the weights between the input layer and hidden layer) and hidden biases (i.e., the thresholds of hidden layer nodes) are randomly generated and remain unchanged throughout the whole training process, while the output weights (i.e., the weights between hidden layer and output layer) are obtained by solving a system of linear matrix equations. The training process of NNRW is non-iterative and thus it can often achieve faster learning speed and comparable accuracy than BP-based neural networks on some issues.

In recent years, NNRW has attracted wide attention and many advances have been made in theory and application [2-6]. According to the differences in the network structure and random degree, NNRW can be divided into two research branches: Extreme learning machine (ELM) [7] and Random vector functional link network (RVFL) [8]. In fact, the training mechanisms of ELM and RVFL are essentially the same [9]. Although NNRW has made lots of breakthroughs in many fields, there are still several fundamental problems that have not been solved such as the model initialization.

Due to the input weights and the hidden bias of NNRW are generated randomly and remain the same during the training process. Therefore, the quality of the model initialization (i.e., the quality of the input weights and hidden biases) has a significant impact on the performance of the NNRW model [10-12].

However, there is no general and appropriate algorithm to help researchers solve the problem of model initialization. In other words, when using NNRW to model different problems, there is no easy way to initialize the network.

*Corresponding author.

978-1-7281-2009-6/$31.00 ©2019 IEEE

Many researchers initialize NNRW with empirical methods, for example, they generate the input weights and hidden biases from the range of (-1, 1) and (0, 1), respectively. However, this method has proven to be not a good choice [10-12]. In recent years, more and more attention has been paid to the initialization problem of NNRW. Researchers have proposed several optimization methods from different perspectives, including Tao et al. [13] have studied the effects of different distributions to initialize ELM on the performance of the model; Cao et al. [14] have conducted similar research based on RVFL; their research has given some suggestions for the selection of the type of initialization distribution. In [15], a method based on the evolutionary algorithm is proposed to improve the initialization quality of NNRW. Wang et al. [16] point out that the orthogonalization of the input weights can improve model performance.

The above works provide some guidelines for choosing appropriate initialization methods for NNRW. However, the authors did not consider the meta features of the experimental datasets such as the distributions of attributes in a dataset. Although they used many datasets in their experiments, it is still difficult for us to understand the relationship between the intrinsic characteristics of the datasets and the initialization of NNRW. This will result in that when we model a new problem, we can only choose a seemingly reasonable initialization method based on the related empirical research results, but not according to the nature of the problem.

Actually, meta features can be used to describe the essential characteristics of a dataset. If there is a certain relationship between the meta features of a dataset and the initialization of NNRW, we can use the meta features to distinguish the differences between different datasets and then choose an appropriate strategy to initialize NNRW for a new problem.

Based on the above analysis, we studied the relationship between meta feature of dataset and initialization of NNRW model in this paper. The meta feature here refers to the attributes' distributions of a dataset. In other words, we studied the relationship between the attributes' distributions of a dataset and the NNRW initialization.

In our experiments, we constructed seven special regression datasets with known attributes' distributions (noted as ***D1***, ***D2***, ***...***, ***D7***) to study the above issues. In each dataset, there are 5000 samples and each sample has three attributes. The label of each sample is obtained by substituting the corresponding attributes' values into a complex function. In ***D1-D3***, all attributes follow the same distribution with the same variances (e.g., in ***D1***, all attributes follow a *Gaussian* distribution with the parameters *mu* = 0 and *sigma* = 0.1). In ***D4-D6***, all attributes follow the same distribution but with different variances (e.g., in ***D4***, the first attribute follows a *Gaussian* distribution with the parameters *mu* = 0 and *sigma* = 0.1, the second attribute follows a *Gaussian* distribution with the parameters *mu* = 0 and *sigma* = 0.3, and the third attribute follows a *Gaussian* distribution with the parameters *mu* = 0 and *sigma* = 0.5). In ***D7***, each attribute follows a specific distribution (i.e., the first attribute follows a *Gaussian* distribution, the second attribute follows a *Gamma* distribution, and the third attribute follows a *Uniform* distribution).

We used three distributions with three variances to initialize NNRW models (both ELM and RVFL were studied), respectively, and then we analyzed the effects of each initialization on the performance of NNRW models and obtained several interesting findings.

The main contributions of this paper can be summarized as:

1) The relationship between the meta features of a dataset and the initialization of NNRW (including both ELM and RVFL) are studied for the first time, which reveals the relationship between the intrinsic characteristics of a dataset and the initialization of NNRW to some extent.

2) Some useful suggestions are provided for researchers to do better NNRW initialization.

3) The theoretical analysis of the relevant experimental results is also given in this paper.

The rest of this paper is organized as follows. In Section II, we give a brief review to NNRW. The details of our experimental settings are given in Section III, including the methods of dataset generation, distribution function selection, and the parameters of NNRW. In Section IV, we give the experimental results and the corresponding analysis. Section V summarizes this paper and shares our future work.

II. Review Of Neural Netorks With Random Weights And Related Works

In this section, we briefly review the Neural Networks with Random Weights (NNRW), including Random Vector Functional Link network (RVFL) and Extreme Learning Machine (ELM). In addition, related research are also reviewed in this part.

A. Random Vector Functional Link Network (RVFL)

Random vector functional link network (RVFL) is a typical NNRW, which was proposed by Pao et al. in the 1990s [8]. The network structure of RVFL with a single hidden layer is shown in Fig. 1, where ω denotes the weights between the input layer and hidden layer (i.e., input weights), b denotes the thresholds of the hidden nodes (i.e., hidden bias), β is the concatenation of the weights between the input layer and output layer and the weights between the hidden layer and output layer (i.e., output weights).

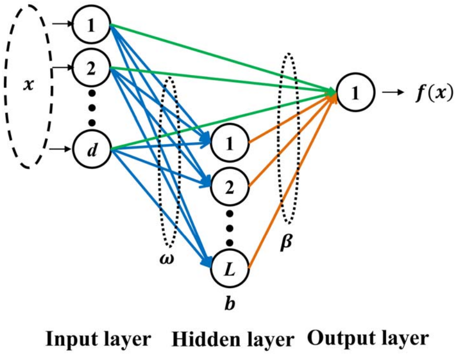

Fig. 1. A typical RVFL with a single hidden layer

In RVFL, ω and b are generated randomly and kept unchanged throughout the training process, while β is obtained analytically.

Given a training dataset $D=\{(x_i,t_i)\,|\,x_i\in R^d, t_i\in R^k\}$, $i=1,2,...,N$, the RVFL shown in Fig. 1 can be modeled as

$$\sum_{i=1}^{L}\beta_i g(\omega_i x+b_i)+\sum_{j=L+1}^{L+d}\beta_j x = y\,, \tag{1}$$

where $g(\cdot)$ denotes the activation function in the hidden layer and y is the prediction values. The optimization objective is

$$\min_{\|\beta\|}\left(\min\sum_{i=1}^{N}\|t_i-y_i\|^2\right) \tag{2}$$

where t is the real values and (2) can be rewrite as

$$H\beta=T \tag{3}$$

where $H=\begin{bmatrix} g(\omega_1\cdot x_1+b_1) & \dots & g(\omega_L\cdot x_1+b_L) & x_1 \\ \vdots & \ddots & \vdots & \vdots \\ g(\omega_1\cdot x_N+b_1) & \cdots & g(\omega_L\cdot x_N+b_L) & x_N \end{bmatrix}_{N\times(L+d)}$,

$\beta=\begin{bmatrix}\beta_1\\ \vdots\\ \beta_{L+d}\end{bmatrix}_{(L+d)\times 1}$, and $T=\begin{bmatrix}t_1\\ \vdots\\ t_N\end{bmatrix}_{N\times 1}$.

The output weights β can be calculated by

$$\beta=H^{+}T\,, \tag{4}$$

where H^{+} is the Moore–Penrose generalized inverse of H.

B. Extreme Learning Machine (ELM)

Extreme learning machine (ELM) was proposed by Huang et al. [7] in 2004. The structure of ELM with a single hidden layer is shown in Fig. 2. Unlike RVFL, there is no direct connection between the input layer and output layer in ELM, and thus ELM is slightly different from RVFL in model training.

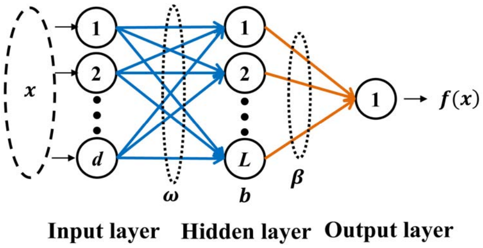

Fig. 2. A typical ELM with a single hidden layer

The ELM shown in Fig. 2 can be modeled as

$$\sum_{i=1}^{L}\beta_i g(\omega_i x+b_i)=y \tag{5}$$

The optimization objective of ELM is same as (2) and we can also rewrite (5) as

$$H\beta=T\,, \tag{6}$$

where $H=\begin{bmatrix} g(\omega_1\cdot x_1+b_1) & \dots & g(\omega_L\cdot x_1+b_L) \\ \vdots & \ddots & \vdots \\ g(\omega_1\cdot x_N+b_1) & \cdots & g(\omega_L\cdot x_N+b_L) \end{bmatrix}_{N\times L}$,

$\beta=\begin{bmatrix}\beta_1\\ \vdots\\ \beta_L\end{bmatrix}_{L\times 1}$, and $T=\begin{bmatrix}t_1\\ \vdots\\ t_N\end{bmatrix}_{N\times 1}$.

The output weight β in ELM can be solved in the same way as (4).

C. Related Works

From the above review of NNRW, it is not difficult to find that the input weights and hidden biases play an important role in the training process of NNRW. Usually, researchers prefer to use the *Uniform* distribution to initialize the NNRW model, where the range of randomly generated input weights is (-1,1) and the range of hidden bias is (0,1). However, several researchers have proved that this empirical method is not the best choice [10-12]. Tao et al. [13] used three distributions (i.e., *Uniform*, *Gaussian*, and *Gamma* distributions) to initialize ELM, respectively, and then compared the classification accuracy of the corresponding models on 30 datasets from UCI machine learning repertory [17]. They observed some interesting phenomena, such as using *Uniform* and *Gamma* distributions with smaller variances to initialize ELM can make the final model have higher classification accuracy while initializing ELM with *Gaussian* distribution is more likely to cause the model overfitting. In [14], Cao et al used ten regression datasets to test the effect of initializing RVFL with different distributions on model performance and also obtained some useful conclusions such as initializing RVFL by a distribution with smaller variance needs less hidden nodes to achieve comparable accuracy with ones having larger variances. In addition, other researchers also proposed some complex algorithms to optimize the initialization of NNRW such as using evolutionary algorithms to select high-quality parameters for NNRW initialization [15], adding orthogonal constraints on initialization parameters to improve the capability of sample structure preserving [16], etc. These algorithms can improve the model performance in some specific applications, but the computational efficiency of NNRW is also reduced because of the introduction of additional computational processes.

III. Experimental Settings

In this session, we give the details of out experimental settings, including the details of datasets, the distribution settings, and the parameters of NNRW models.

A. Datasets Generating

In our experiments, we constructed seven special datasets (denoted as ***D1***, ***D2***, ***D3***, ***D4***, ***D5***, ***D6***, and ***D7***) to study the relationship between the attributes' distributions of a dataset and the initialization of NNRW. The details of the datasets are given in the Table I. In each dataset, there are 5000 samples and each sample has three attributes (denoted as ***A1***, ***A2***, and ***A3***). The label of each sample is obtained by substituting the corresponding attribute values into the following *Sphere* function:

$$f=\sum_{i=1}^{D}x_i^2\ , \tag{7}$$

It is noted that the initial range of the variables of the *Sphere* function is $[-100,100]$.

TABLE I. THE DETAILS OF THE EXPERIMENTAL DATASETS

Dataset	Distribution	Parameters
D1	***A1***, ***A2***, and ***A3*** follow the *Gaussian* distribution.	μ=0.0, σ=0.5
D2	***A1***, ***A2***, and ***A3*** follow the *Gamma* distribution.	$k=9.0,\theta=0.4$
D3	***A1***, ***A2***, and ***A3*** follow the *Uniform* distribution.	$a=-1,b=1$
D4	***A1***, ***A2***, and ***A3*** follow the *Gaussian* distribution.	$A1\rightarrow(\mu_1=0.0,\sigma_1=0.1)$ $A2\rightarrow(\mu_2=0.0,\sigma_2=0.3)$ $A3\rightarrow(\mu_3=0.0,\sigma_3=0.5)$
D5	***A1***, ***A2***, and ***A3*** follow the *Gamma* distribution.	$A1\rightarrow(k_1=9.0,\theta_1=0.2)$ $A2\rightarrow(k_2=9.0,\theta_2=0.4)$ $A3\rightarrow(k_3=9.0,\theta_3=0.6)$
D6	***A1***, ***A2***, and ***A3*** follow the *Uniform* distribution.	$A1\rightarrow(a_1=0.0,b_1=1.0)$ $A2\rightarrow(a_2=-1.0,b_2=1.0)$ $A3\rightarrow(a_3=-2.0,b_3=2.0)$
D7	***A1*** follows the *Gaussian* distribution, ***A2*** follows the *Gamma* distribution, and ***A3*** follows the *Uniform* distribution.	$A1\rightarrow(\mu=0.0,\sigma=0.5)$ $A2\rightarrow(k=9.0,\theta=0.2)$ $A3\rightarrow(a=0.0,b=1.0)$

B. Distributions for the Initialization of NNRW

In this paper, we used three common distributions (i.e., *Uniform*, *Gaussian*, and *Gamma* distributions) to initialize the NNRW and study the relationship between the attributes' distributions of a dataset and the model initialization. For each distribution, there are three different parameter pairs. The details of these distributions are shown in the Table II.

TABLE II. THE DISTRIBUTIONS FOR THE INITIALIZATION OF NNRW

Distribution	Parameters
Uniform	$a=0,b=1$
	$a=-1,b=1$
	$a=-2,b=2$
Gamma	$k=9,\theta=0.2$
	$k=9,\theta=0.4$
	$k=9,\theta=0.6$
Gaussian	$\mu=0,\sigma=0.05$
	$\mu=0,\sigma=0.2$
	$\mu=0,\sigma=0.35$

C. Parameters for NNRW

- The type of NNRW. Both the ELM and RVFL are tested in our experiments.
- The activation function. Three common non-linear functions (i.e., *Sigmoid function*, *Radial basis function*, and *Triangular basis transfer function*) are used as the activation function of NNRW, respectively.
- The number of hidden layer nodes is set from 1 to 100.

IV. EXPERIEMENTAL RESULTS AND ANALYSIS

Our experiments are conducted in the MATLAB R2016b environment on the same Windows 7 OS with Intel i7-6700 3.4 GHz CPU and 32 GB RAM. The result of each experiment is the average of 50 experiments. The experimental results show that the experimental observations are insensitive to activation function and the type of NNRW. Due to the limitation of the number of pages, this paper only shows the experimental results of ELM with the *Sigmoid* function, i.e., $g(\omega,x,b)=1/(1+\exp(-(\omega\bullet x+b)))$. Figs. 3-12 show the training and testing RMSE (Root Mean Square Error) of the relevant models on the dataset ***D1-D7***. It is noted that the key local areas of the figures are magnified.

A. Experimental Resutls and Analysis

Figs. 3-9 show the effect of initializing ELM with different distributions on the model performance when the attributes in a dataset follow the same distribution with the same variances (i.e., ***D1***, ***D2***, and ***D3***).

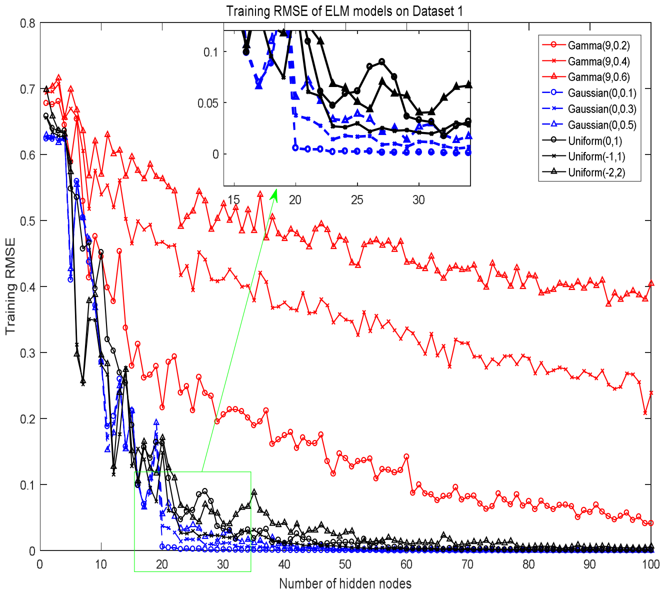

Fig. 3 Training RMSE of ELM models on ***D1***

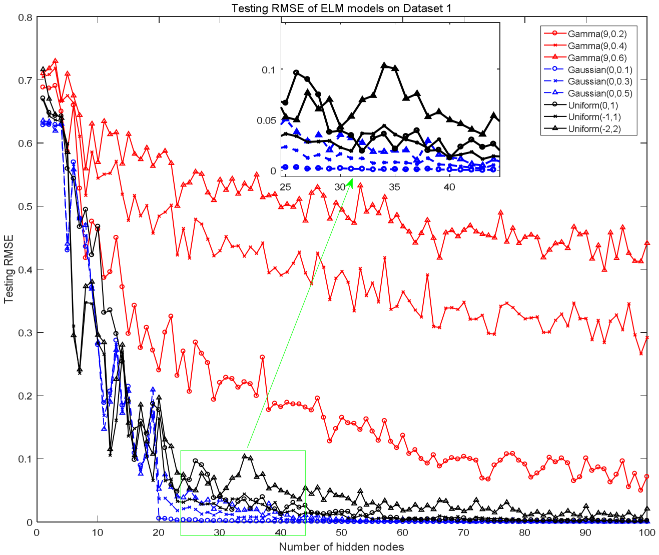

Fig. 4 Testing RMSE of ELM models on ***D1***

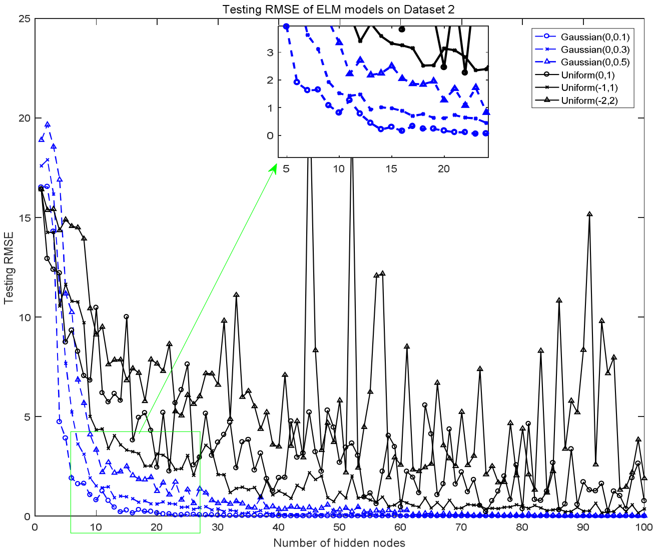

Fig. 7 Testing RMSE of ELM models on ***D2***

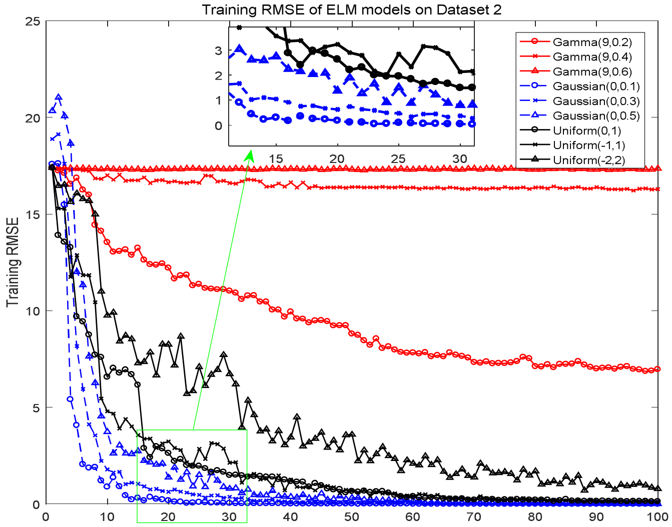

Fig. 5 Training RMSE of ELM models on ***D2***

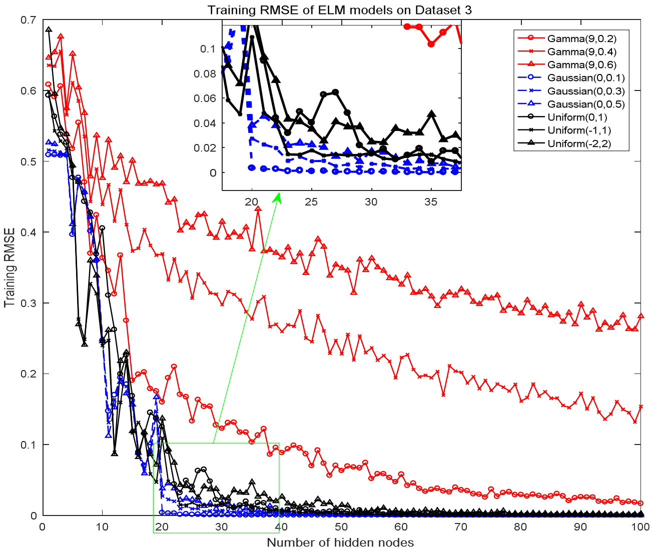

Fig. 8 Training RMSE of ELM models on ***D3***

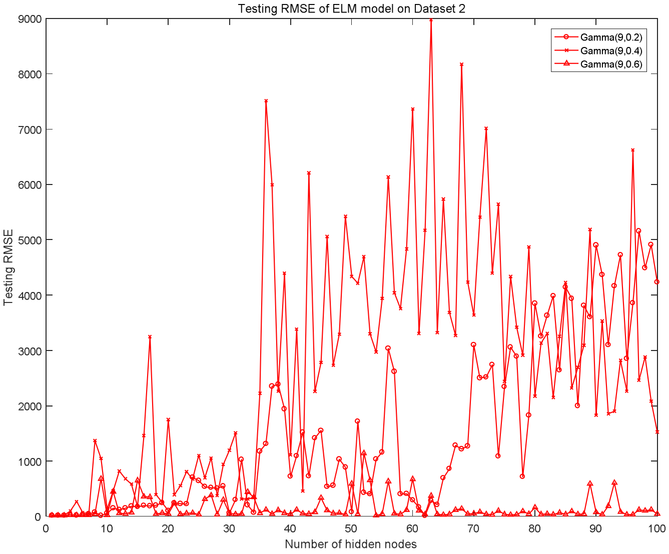

Fig. 6 Testing RMSE of ELM models on ***D2*** (*Gamma*)

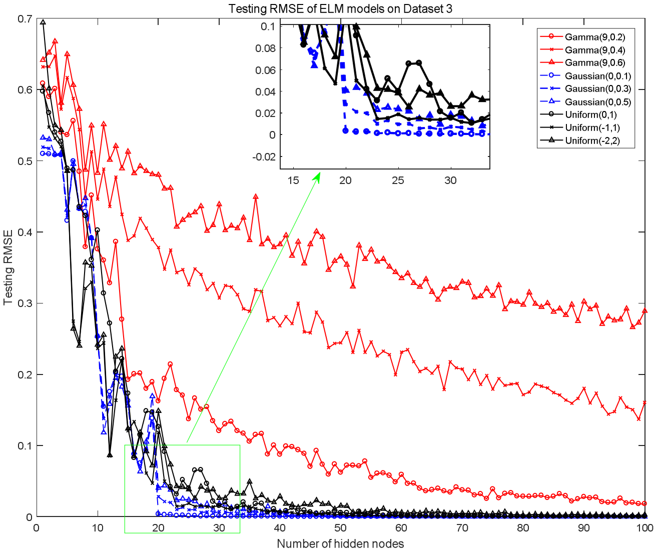

Fig. 9 Testing RMSE of ELM models on ***D3***

From Figs. 3-9, we can observe that:

1) In all the figures, the blue lines are always faster and closer to the bottom than the black and red lines, which implies that the blue lines can converge faster and achieve lower errors. Therefore, we can infer that: Regardless of the attributes' distributions of a dataset, initializing ELM with Gaussian distribution can make the model converge faster than ones with Gamma and Uniform distributions.

2) In the training RMSE figures, except for very few cases, the black and red lines will eventually be close to the bottom, but they do not fall as fast as the blue lines, which implies that: Using Gamma and Uniform distributions to initialize ELM often requires more hidden layer nodes to achieve comparable accuracy as ones using Gaussian distribution.

3) From the training RMSE curve in Fig. 5, we can observe that the convergence rates of red lines are much slower than the blue and black lines, and in the testing RMSE curves (i.e., Figs. 6-7), the red lines have sharp fluctuations and the error values are much larger than that of the blue and black lines. We know that all the attributes of **D2** follow Gamma distribution. Therefore, we can infer that: If the attributes of a dataset follow Gamma distribution, then using Gamma distribution to initialize ELM may cause the model over-fitting.

The effect of initializing ELM with different distributions on the model performance when the attributes in a dataset follow the same distribution but with different variances (i.e., ***D4***, ***D5***, and ***D6***) was also studied. We found that the experimental results show a similar phenomenon to the above. Specifically, Firstly, initializing ELM with *Gaussian* distribution can always make the model have a faster convergence rate than ones using *Gamma* and *Uniform* distributions. Secondly, using *Gaussian* distribution to initialize ELM often requires less hidden layer nodes to achieve comparable accuracy as ones using *Gamma* and *Uniform* distributions. Thirdly, when the attributes of a dataset that follow *Gamma* distribution, it is not a good idea to use *Gamma* distribution to initialize ELM.

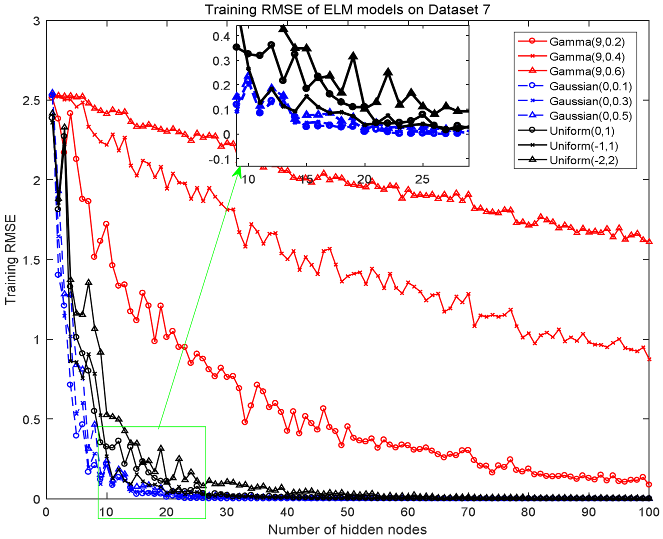

Fig. 10 Training RMSE of ELM models on ***D7***

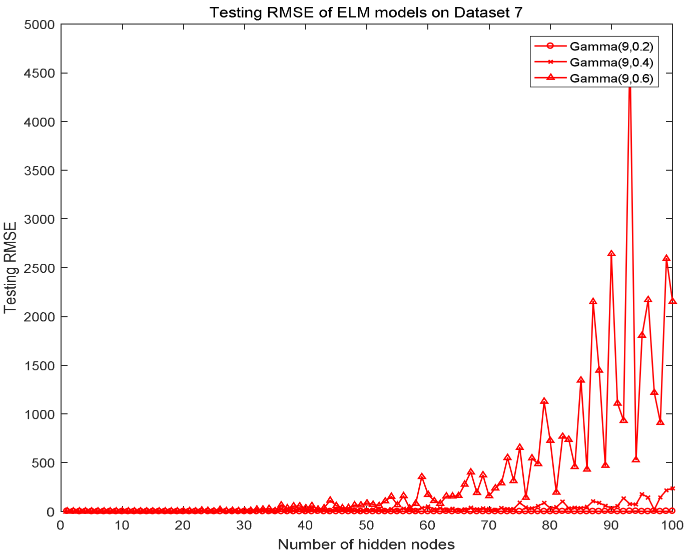

Fig. 11 Testing RMSE of ELM models on ***D7*** (*Gamma*)

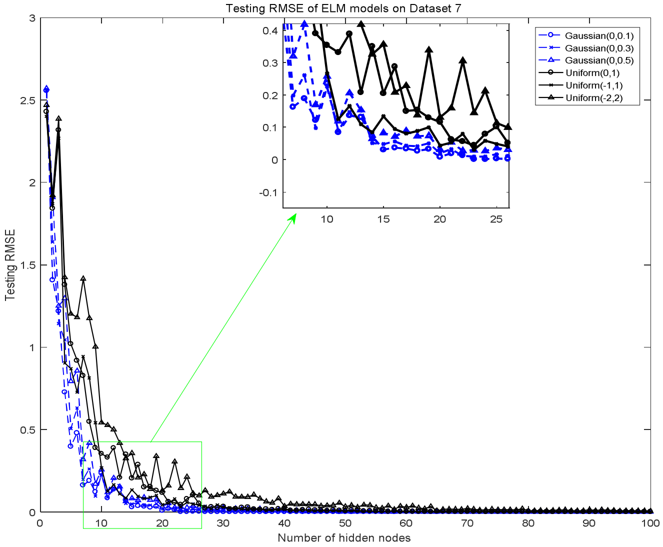

Fig. 12 Testing RMSE of ELM models on ***D7***

Figs. 10-12 shows the effect of initializing ELM with different distributions on the model performance when the attributes of a dataset follow different distributions (i.e., ***D7***).

From Figs. 10-12, we can observe that using *Gaussian* distribution to initialize ELM can make the model converge faster than the one using *Gamma* and *Uniform* distributions. Since the second attribute of the dataset follows *Gamma* distribution, initializing ELM with *Gamma* distribution can easily lead to the model over-fitting.

B. Theoretical explanations

Some theoretical explanations about the above observations are given as follows:

Assume there is an ELM model denoted as $y=\sum_{i=1}^{L}\beta_i g(\omega_i x+b_i)$, the input weight ω_i and hidden bias b_i are generated independently and identically distributed with a distribution μ . Here we consider three cases, that is, the μ is a *Gaussian* distribution, a *Uniform* distribution, or a *Gamma* distribution with a certain variance σ^2 .

The reason why using the *Gamma* distribution to initialize the NNRW is not a good choice is that the *Gamma* distribution may make the variance of $\omega_i x + b_i$ larger. To simplify the proof, here we consider a simple case where x is one-dimensional data, and then

$$\begin{aligned} Var(\omega_i x + b_i) &= Var(\omega_i x) + Var(b_i) \\ &= E(\omega_i x)^2 - E(\omega_i^2 x^2) + Var(b_i) \\ &= E(\omega_i)^2 E(x)^2 - E(\omega_i^2)E(x^2) + Var(b_i) \\ &= E(\omega_i)^2 E(x)^2 - (Var(\omega_i) + E(\omega_i)^2)E(x^2) + Var(b_i) \\ &= Var(x)E(\omega_i)^2 - Var(\omega_i)E(x)^2 + Var(b_i). \end{aligned} \tag{8}$$

As we know, *Gamma* distribution always has a positive mean, which means that (8) is larger when ω_i follows a *Gamma* distribution than a *Uniform* or *Gaussian* distribution with mean 0 and the same variance. If $\omega_i x + b_i$ has a larger variance, $g(\omega_i x + b_i)$ is highly likely to have a larger variance, too. This is also an explanation why the *Gamma* distribution may lead to an overfitting of the model: It creates a model with a larger variance. Also, (8) implies that a larger variance of μ will lead to a larger variance of $\omega_i x + b_i$, so it is better to choose μ with a small variance.

The calculation of β_i follows the method of least squares. We notice the fact that the method of least squares for linear model gives the best regression result on normally distributed statistics in the sense that the regression model is unbiased and has the minimal residual error. If the statistics are not normally distributed, generally it does not hold any more. To determine which distribution μ results in a better regression result, a natural way is to compare the difference of the distribution of $g(\omega_i x + b_i)$ and the *Gaussian* distribution with the same mean and variance with different μ choices. It is natural to predict that the distribution of $g(\omega_i x + b_i)$ is more similar to a *Gaussian* distribution if we choose μ to be normal, especially when the statistics x also follows a *Gaussian* distribution.

V. Conclusions

In this paper, we studied the relationship between the attributes' distributions of a dataset and the initialization of NNRW, and obtained some interesting observations as follows.

1) Initialize NNRW with *Gaussian* distribution can make the model have faster convergence rate than ones with *Gamma* and *Uniform* distribution in our experiments. And this phenomenon is insensitive to the attributes' distributions of a dataset.

2) if one or more attributes in a dataset that follows *Gamma* distribution, then using *Gamma* distribution to initialize NNRW may result in a slow convergence and easy overfitting of the model.

3) Usually, using a specific distribution with smaller variances can make the NNRW model achieve faster convergence rate and better generalization performance than ones with larger variances. This phenomenon is independent of the attributes' distributions of a dataset.

4) The above observations are insensitive to the type of activation functions and the type of NNRW.

5) In addition, changes in the variance of the attributes' distributions of a dataset do not affect the above conclusions.

The above experimental results provide useful guidelines for researchers to choose an appropriate initialization method for NNRW. In the actual use of the above guidelines, one needs to get a general idea of the attributes' distributions of the dataset in advance. Of course, sometimes this is very difficult. One suggestion is to use the prior knowledge of the problem or some assistant tools such as Minitab [18] to analyze the relevant information of the dataset. In addition, expectation–maximization (EM) algorithm [19] is also a choice for determining the distribution of variables.

This paper is an initial study on the relationship between meta features of a dataset and the initialization of NNRW. Some limitations are given here: Firstly, only three common distributions are considered in the paper. In fact, there are still many other distributions that can be studied. Secondly, the datasets used in the paper are not complex enough, and there is no experiment based on real datasets. Thirdly, the paper only focuses on a special type of meta features of a dataset (i.e., the attributes' distributions). Actually, there are many meta features can be used to describe the intrinsic characteristics of a dataset such as the degree of imbalance and the training set size. In the future, we will study the above issues to get more comprehensive and solid conclusions.

Acknowledgment

We really thank the editor and anonymous reviewers for their invaluable suggestions to help us improve this paper. This work was supported by the National Natural Science Foundation of China under Grant nos. 61672358 and 61836005.

References

[1] X. Z. Wang, and W. P. Cao, "Non-iterative approaches in training feed-forward neural networks and their applications," Soft Computing, vol. 22, 2018, pp. 3473-3476.

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

[3] Y. Q. Chen, C. Y. Hu, B. Hu, L. S. Hu, H. Yu, and C. Y. Miao, "Inferring Cognitive Wellness from Motor Patterns," IEEE Trans. Knowl. Data Eng., vol. 30, no. 12, 2018, pp. 2340-2353.

[4] C. A. S. da Silva and R. A. Krohling, "Semi-Supervised Online Elastic Extreme Learning Machine for Data Classification," In IJCNN, IEEE, 2018, pp. 1-8.

[5] P. A. Henríquez and G.A. Ruz, "Twitter Sentiment Classification Based on Deep Random Vector Functional Link," In IJCNN, IEEE, 2018, pp. 1-6.

[6] P. Dai, F. Gwadry-Sridhar, M. Bauer, M. Borrie, and X. Teng, “Healthy Cognitive Aging: A Hybrid Random Vector Functional-Link Model for the Analysis of Alzheimer's Disease,” In AAAI, 2017, pp. 4567–4573.

[7] G. B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: a new learning scheme of feedforward neural networks,” in IJCNN, IEEE, 2004, pp. 985-990.

[8] Y. H. Pao and Y. Takefuji, “Functional-link net computing: theory, system architecture, and functionalities,” Computer, vol. 25, no. 5, 1992, pp. 76-79.

[9] W. P. Cao, X. Z. Wang, Z. Ming, and J. Z. Gao, “A review on neural networks with random weights,” Neurocomputing, vol. 275, 2018, pp. 278-287.

[10] L. Zhang and P. N. Suganthan, “A comprehensive evaluation of random vector functional link networks,” Inf. Sci., vol. 367, 2016, pp. 1094-1105.

[11] M. Li and D. H. Wang, “Insights into randomized algorithms for neural networks: practical issues and common pitfalls,” Inf. Sci., vol. 382, 2017, pp. 170-178.

[12] W. P. Cao, J. Z. Gao, Z. Ming, and S. B. Cai, “Some Tricks in Parameter Selection for Extreme Learning Machine,” In IOP Conference Series: Materials Science and Engineering, vol. 261, no. 1, IOP Publishing, 2017, p. 012002.

[13] X. Tao, X. Zhou, Y. L. He, and R. A. R. Ashfaq, “Impact of variances of random weights and biases on extreme learning machine,” Journal of Software, vol. 11, no. 5, 2016, pp. 440-454.

[14] W. P. Cao, J. Z. Gao, Z. Ming, S. B. Cai, and H. Zheng, “Impact of Probability Distribution Selection on RVFL Performance,” In SmartCom, Springer, 2017, pp. 114-124.

[15] M. Eshtay, H. Faris, and N. Obeid, “Improving extreme learning machine by competitive swarm optimization and its application for medical diagnosis problems,” Expert Syst. Appl., vol. 104, 2018, pp. 134-152.

[16] W. H. Wang, and X. Y. Liu, “The selection of input weights of extreme learning machine: A sample structure preserving point of view,” Neurocomputing, vol. 261, 2017, pp. 28-36.

[17] M. Lichman, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.

[18] Minitab Inc. MINITAB statistical software, release 14 for Windows. State College, PA. 2005.

[19] T. K. Moon, “The expectation-maximization algorithm,” IEEE Signal Process. Mag., vol. 13, no. 6, 1996, pp. 47-60.