Ensemble Neural Networks with Random Weights for Classification Problems

Ye Liu
College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
ly@szu.edu.cn

Weipeng Cao*
College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.
caoweipeng@szu.edu.cn
(*Corresponding author)

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

Qiang Wang
Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
wangq8@sustech.edu.cn

Jiyong Zhang
School of Automation, Hangzhou Dianzi University, Hangzhou, China
jzhang@hdu.edu.cn

Zhiwu Xu
College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. xuzhiwu @szu.edu.cn

ABSTRACT

To improve the prediction accuracy and stability of neural networks with random weights (NNRWs), we propose a novel ensemble NNRWs (E-NNRW) in this paper, which initializes its base learners by different distributions to improve their diversity. The final prediction results of the E-NNRW model are determined by these base learners through a voting mechanism, which minimizes the specific "blind zone" of a single learner, thus achieving higher prediction accuracy and better stability. Taking the random vector functional link network (RVFL), one of the most representative algorithms in NNRWs, as an example, we fully evaluate the performance of the proposed algorithm on nine benchmark classification problems. Extensive experimental results fully demonstrate the effectiveness of our method.

CCS CONCEPTS

• **Computing methodologies**; • **Artificial intelligence**; • **Philosophical/theoretical foundations of artificial intelligence**;

KEYWORDS

Neural networks with random weights, random vector functional link network, ensemble learning, randomized neural networks

ACM Reference Format:
Ye Liu, Weipeng Cao*, Zhong Ming, Qiang Wang, Jiyong Zhang, and Zhiwu Xu. 2020. Ensemble Neural Networks with Random Weights for Classification Problems. In *2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2020), December 24–26, 2020, Sanya, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3446132.3446147

ACAI 2020, December 24–26, 2020, Sanya, China

ACM ISBN 978-1-4503-8811-5/20/12…$15.00
https://doi.org/10.1145/3446132.3446147

1 INTRODUCTION

Traditional neural network algorithms, including deep learning, have strong feature representation and learning capabilities, which have made breakthroughs in many fields, especially computer vision [1], machine translation [2], etc. However, these algorithms also suffer from several notorious weaknesses, such as low training efficiency, many hyperparameters and difficulty in setting their values, and extremely high requirements for hardware computing resources [3]. These shortcomings limit the application of traditional neural network algorithms in many Internet of things (IoT) scenarios because the computing power of the hardware in these scenarios is relatively limited.

To avoid the above-mentioned shortcomings of traditional neural networks, neural networks with random weights (NNRWs) were proposed [4][5]. The biggest difference between NNRWs and traditional neural networks is that NNRWs adopt a non-iterative training mechanism. Specifically, NNRWs do not need to adjust all parameters of the network iteratively through the gradient descent method like traditional neural networks. NNRWs assume that some parameters in the network can remain unchanged after being initialized according to certain rules. For example, for an NNRW with a single hidden layer, its input weights (i.e., the weights between the input layer and the hidden layer) and hidden biases (i.e., the thresholds of the hidden nodes) can be randomly assigned by certain rules, we only need to calculate its output weight (i.e., the weights between the hidden layer and the output layer) by using the least square method to complete the model training.

The above-mentioned non-iterative training mechanism brings the NNRWs extremely fast learning speed and good generalization ability, which has attracted extensive attention in recent years, and various improved algorithms and applications have been proposed. The related representative work includes random vector functional link network (RVFL) [6], extreme learning machine (ELM) [7], stochastic configuration network (SCN) [8], broad learning system (BLS) [9], etc. The training mechanism of these algorithms is the same, while there are some differences in details. For example, in order to fuse the low-level features and extracted high-level features to improve the decision-making ability of the model, RVFL connects the input layer and the output layer directly. For faster training speed, ELM removes the links between the input layer and the output layer. To better initialize the input weights and hidden biases, SCN adopts a supervisory random mechanism to initialize its model. BLS improves its feature extraction ability by adding an extra feature enhancement layer.

Although the universal approximation ability of NNRWs has been widely proved [8][10][11], the premise of this conclusion is that the number of hidden layer nodes in the network tends to be infinite. However, this premise is difficult to be satisfied in practical applications. Therefore, it is difficult to guarantee the stability of the model due to the fact that the random parameters in such algorithms remain unchanged throughout the model training process [12].

To solve this problem, some researchers have proposed to use the ensemble learning mechanism to improve the stability of NNRWs. The basic idea of this kind of work is that since a single model has a specific "blind zone" due to the randomness of its parameters, one can use multiple base learners to jointly make decisions to reduce the blind zone of the model as much as possible. In this way, the prediction ability and robustness of the final model can be improved. The representative work includes: Zhai JH et al. [13] transform imbalanced problems into corresponding balanced ones by manipulating the distribution of the classes in the way of oversampling, and then use different data blocks to train ELM models and realize ensemble learning based on the voting mechanism. In [14], the authors use different activation functions to improve the diversity of base learners (online sequential ELM) and then use the voting mechanism to make decisions. Similar work has been done in [15] and [16]. Wang DH et al. [17] use SCN as the base learner and evaluate the output weights of the ensemble SCN models through the negative correlation learning strategy (NCL).

At present, most of the existing work still use the uniform distribution to initialize their base learners. However, our previous study found that initializing NNRWs with different distributions has a significant impact on model performance [12][18]. In other words, the distribution most suitable for the initialization of NNRWs varies from dataset to dataset. However, most of the NNRWs only use the uniform distribution to initialize their input weights and hidden biases, which leads to the poor generalization ability of these models in practical applications.

Inspired by this phenomenon and the ensemble learning mechanism, this paper proposes a novel ensemble NNRW framework (E-NNRW), in which each base learner is initialized with a specific distribution (i.e., the uniform distribution, Gaussian distribution, Gamma distribution). The final prediction results of the model are determined by the voting mechanism of these base learners, so the misjudgment caused by the blind zone of a single learner is effectively reduced.

The contribution of this paper can be summarized as follows:

(1) We propose an ensemble learning framework based on different initialization for NNRWs, which is easy to implement and can effectively improve the prediction accuracy and stability of the model.

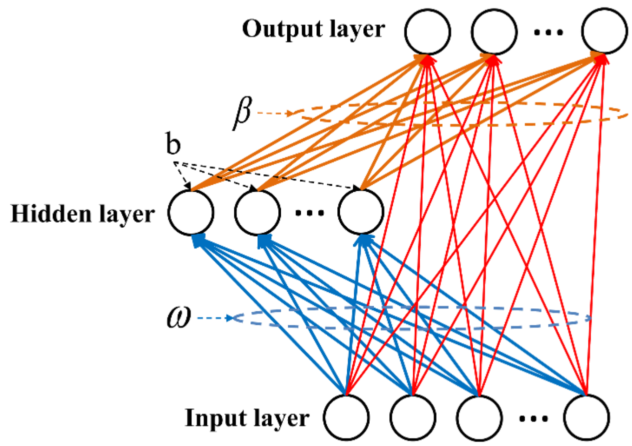

Figure 1: The network structure of RVFL

(2) The idea of E-NNRW can be directly transferred to various mainstream NNRW algorithms, such as RVFL, ELM, SCN, and BLS, to improve their robustness and generalization ability.

(3) Taking E-RVFL as an example, we have fully verified the effectiveness of the proposed method on nine benchmark classification problems.

The rest of this paper is organized as follows. In Sec. 2, we take RVFL as an example to briefly introduce the training mechanism of NNRWs. We introduce the details of the proposed E-NNRW algorithm in Sec. 3. We give the configuration and results of the experiment in Sec. 4. In Sec. 5, we conclude this paper.

2 PRELIMINARIES

In this section, we take RVFL as an example [19] to briefly introduce the training mechanism of NNRWs. Note that, as described in Sec. 1, the training mechanism of other NNRW algorithms such as ELM, SCN, and BLS is similar to RVFL.

As shown in Fig. 1, this is a network structure of RVFL with a single hidden layer. In Fig. 1, $\boldsymbol{\omega}$ refers to the input weights and $\boldsymbol{b}$ refers to the hidden biases, which both are generated from a given range (i.e., [-1, 1]) under a uniform distribution. The output weights $\boldsymbol{\beta}$ are calculated by the least square method. The training mechanism of RVFL can be summarized as Algorithm 1

From algorithm 1, one can easily infer that the training mechanism of the RVFL is non-iterative. Compared with those traditional neural networks with the iterative training mechanism, RVFL can achieve faster training speed when the scale of data sets is relatively small [4].

However, one can also observe that since $\boldsymbol{\omega}$ and $\boldsymbol{b}$ are randomly generated when the model is initialized, and remain unchanged throughout the training process of the model, if their quality is not high enough, it will have a significant negative impact on the model performance. In fact, we have studied the influence of the generation range and the obeyed distribution of $\boldsymbol{\omega}$ and $\boldsymbol{b}$ on the model performance and gave some valuable guidelines to help researchers better initialize the NNRW model [12][18][20].

However, the above guidelines are for specific data sets and a single model. To further enhance the generalization ability and

Algorithm 1 RVFL algorithm

Input: Given a training data set $\boldsymbol{D} = \{\boldsymbol{X}, \boldsymbol{T}\} \in \boldsymbol{R}^{(\boldsymbol{d}+\boldsymbol{m})\times \boldsymbol{N}}$, where $\boldsymbol{d}$ is the dimension of the input data, $\boldsymbol{m}$ is the class number for classification problems, and $\boldsymbol{N}$ is the number of samples. Set the number of hidden nodes in RVFL to $\boldsymbol{L}$ and the activation function to $\boldsymbol{g}(\cdot)$.
Output: The parameters of the RVFL model.
Step 1. Randomly generate the values of $\boldsymbol{\omega}$ and $\boldsymbol{b}$ from a given range (i.e., [-1, 1]) under a uniform distribution.
Step 2. Calculate the output of the hidden layer:
$H_0 = \sum_{i=1}^{L} g(\omega_i \cdot x_j + b_i),\ j = 1, .., N \quad (1)$
where $\omega_i \cdot x_j$ refers to the inner product of ω_i and x_j.
Step 3. Joint the output of the hidden layer with the original feature matrix:
$H = [H_0,\ X] \quad (2)$
Step 4. Calculate the output weights β:
When $N < L$,
$\beta = H^T(HH^T)^{-1}T \quad (3)$
When $N > L$,
$\beta = (H^TH)^{-1}H^TT \quad (4)$
Step 5. Return $\boldsymbol{\omega}$, $\boldsymbol{b}$, and $\boldsymbol{\beta}$.

robustness of the model, we propose a new algorithm based on the ensemble learning mechanism in the next section.

3 ENSEMBLE NEURAL NETWORKS WITH RANDOM WEIGHTS (E-NNRW)

In this section, we introduce the proposed ensemble learning framework for neural networks with random weights (E-NNRW).

Since the input weights and hidden biases of the model remain unchanged after initialization, our basic idea is to initialize NNRWs with different distributions to get diverse base learners. When a testing sample is given, we use the voting mechanism to predict its label. Each model has its specific "blind zone", and this ensemble method can effectively reduce the "blind zone" of the final model, and can obtain higher prediction accuracy and stability than a single model.

Specifically, assuming that there are three base learners, we use three common distribution functions (i.e., uniform distribution, Gaussian distribution, and Gamma distribution) to generate input weights and hidden biases for them. Since these random parameters remain unchanged during the model training process, it can be guaranteed that these base learners can map the original data features from different perspectives. The training process of the base learners is consistent with the original NNRWs, which naturally inherits the advantages of the fast training speed and good generalization ability. We can summarize the E-NNRW algorithm as shown in algorithm 2
The above learning pipeline can be summarized as Fig. 2

4 EXPERIMENTS AND ANALYSIS

In this section, we evaluate the performance of the proposed algorithm on nine benchmark classification problems from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/index.php).

Algorithm 2 E-NNRW algorithm

Input: Given a training data set $\boldsymbol{D} = \{\boldsymbol{X}, \boldsymbol{T}\} \in \boldsymbol{R}^{(\boldsymbol{d}+\boldsymbol{m})\times \boldsymbol{N}}$, where $\boldsymbol{d}$ is the dimension of the input data, $\boldsymbol{m}$ is the class number for classification problems, and $\boldsymbol{N}$ is the number of samples. Set the number of the base learners to $\boldsymbol{K}$, the number of hidden nodes in each learner to $\boldsymbol{L}$, and the activation function of the base learners to $\boldsymbol{g}(\cdot)$.
Output: The parameters of the E-NNRW model.
Training phase:
Step 1. Randomly generate the values of $\boldsymbol{\omega}$ and $\boldsymbol{b}$ from a given range under different distributions (i.e., the uniform distribution, Gaussian distribution, and Gamma distribution) for the $\boldsymbol{K}$ base learners.
Step 2. Train the base learners according to the training mechanism of NNRWs. For example, for the case where the base learner is RVFL, perform steps 2-5 of Algorithm 1 to complete the training of each base RVFL model to obtain the E-RVFL model.
Step 3. Return all the parameters of the E-NNRW model.
Testing phase:
Step 4. Given a testing sample, use each base learner in the E-NNRW model to predict its label, and then use a voting mechanism (i.e., the principle of majority priority) to get its final label.

Due to the space limitation, here we use RVFL, one of the most representative NNRW, as the base learner to train the E-RVFL model. Note that for other NNRW algorithms, such as ELM, SCN, and BLS, it can also be used as the base learner to train the corresponding ensemble model, and the experimental phenomena are similar to the E-RVFL shown in this study.

4.1 Details of Data Sets and Experimental Parameters

Without loss of generality, we chose six binary classification problems and three multi-classification problems to evaluate the effectiveness of the E-RVFL. The details of these data sets are shown in Table 1

In our experiments, in order to avoid the negative impact on the model due to the different dimensions of the attributes, we performed the Min-Max normalization process on the attributes of each data set, the formula is as follows:

$$x_i' = \frac{x_i - min\,(x_i)}{max\,(x_i) - min\,(x_i)},\ i = 1, 2, \cdots, d \tag{1}$$

where x_i refers to the i-th attribute.

After preprocessing, each data set was divided into the training set and the testing set according to 7:3. For simplicity, we set the number of the base learner to 3, the number of the hidden layer nodes in each base learner to 20, and the activation function to the Sigmoid function as follows:

$$g\,(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

For these three base learners, we chose the uniform distribution, Gaussian distribution, and Gamma distribution to initialize them respectively. For the uniform distribution, its parameter is $U(-1,\ 1)$;

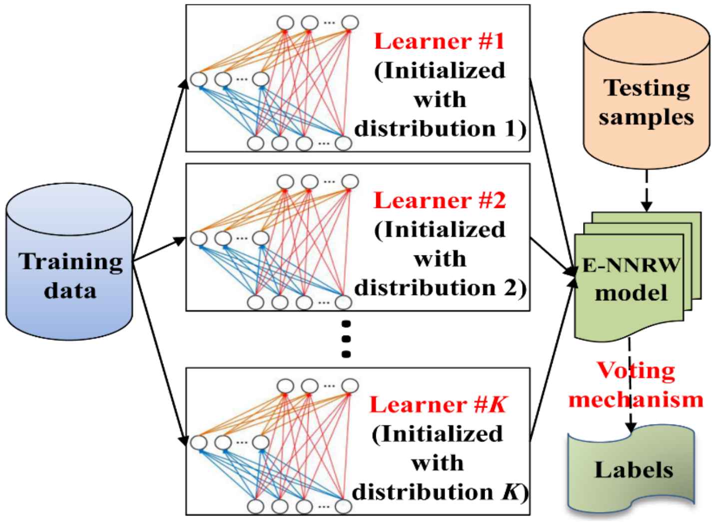

Figure 2: The learning pipeline of E-NNRW

Table 1: The details of the experimental data sets

Dataset	Attributes	Classes	Samples number
Pima	8	2	768
Credit	6	2	690
Spambase	57	2	458
Page	10	3	532
Magic	10	2	19020
AU1-21	20	2	1000
Wilt	5	2	500
Abalone	7	3	1379
Wine_QW	11	3	4535

for the Gaussian distribution, its parameter is $N(0,\ 0.05)$; for the Gamma distribution, its parameter is $Ga(9,\ 0.1)$.

All the experiments were conducted in the MATLAB R2014a environment on the same Windows 10 OS machine with Intel Core i5-5300U CPU and 8 GB RAM.

4.2 Experimental Results and Analysis

According to the parameter settings in 4.1, we evaluated the performance of E-RVFL and RVFL on the above nine classification problems. The RVFL here adopted the default initialization method, that is, the randomly generated input weights and hidden biases obey the uniform distribution. The main performance indicators that this study focuses on include: training accuracy, testing accuracy, and training time of the model. The details of the experimental results are shown in Table 2

Note that the experimental results in Table 2 are the average of the results obtained after 20 independent experiments on each data set. The best experimental results have been bolded.

It can be observed from Table 2 that our proposed E-RVFL can achieve higher training accuracy and testing accuracy than the single RVFL on all data sets, which means that E-RVFL has better predictive ability than RVFL. In other words, the E-RVFL model can achieve better generalization ability than a single RVFL model.

At the same time, one can also observe that the training time of the E-RVFL model is longer than that of the RVFL model. This phenomenon is because the ensemble learning method involves the training of multiple classifiers, and the training time of the ensemble model increases linearly as the number of base learners increases.

Remark: It is a simple and effective way to improve the generalization ability of the NNRW model by using different distribution functions to initialize and construct the ensemble NNRW. This method improves the diversity of base learners through different distributions and uses the voting mechanism to determine the final output, which can minimize the blind zone of a single model, thereby obtaining more accurate prediction results.

5 CONCLUSIONS

To improve the generalization ability and robustness of NNRWs, this paper designs a novel ensemble learning framework for them, which we call E-NNRW. E-NNRW uses different distribution functions to initialize the base learners to increase the diversity of base

Table 2: The performance comparison of the E-RVFL and RVFL

Datasets	Algorithm	Accuracy		Training time
		Training	Testing	
Pima	RVFL	0.5789	0.5169	0.0156
	E-RVFL	0.5878	**0.5339**	0.0469
Credit	RVFL	0.5580	0.5111	0.0141
	E-RVFL	0.5652	**0.5362**	0.0539
Spambase	RVFL	0.8981	0.7217	0.0141
	E-RVFL	0.9111	**0.8042**	0.0773
Page	RVFL	0.9202	0.9012	0.0116
	E-RVFL	0.9326	**0.9130**	0.0594
Magic	RVFL	0.6771	0.6615	0.1031
	E-RVFL	0.6863	**0.6798**	0.3055
AU1-21	RVFL	0.5031	0.4640	0.0186
	E-RVFL	0.5171	**0.5000**	0.0570
Wilt	RVFL	0.7820	0.7173	0.0163
	E-RVFL	0.7886	**0.7333**	0.0422
Abalone	RVFL	0.6173	0.6093	0.0178
	E-RVFL	0.6177	**0.6119**	0.0516
Wine_QW	RVFL	0.5925	0.5786	0.0148
	E-RVFL	0.5941	**0.5850**	0.0672

models, and its final decision result is jointly determined by these base models using the voting mechanism. E-NNRW minimizes the blind zone of a single NNRW model and effectively improves its predictive ability. We use RVFL as the base learner to fully demonstrate the effectiveness of the proposed method on nine benchmark classification problems. Moreover, the idea of E-NNRW can be directly applied to improve the performance of other NNRW algorithms, such as ELM, SCN, and BLS.

In the future, we will consider improving the diversity of the base NNRW model from the perspective of the statistical characteristics of the training data, which is expected to further enhance the generalization ability and robustness of E-NNRW

ACKNOWLEDGMENTS

This work was supported by the Opening Project of Shanghai Trusted Industrial Control Platform (TICPSH202003008-ZC) and the National Natural Science Foundation of China (61672358, 61836005).

REFERENCES

[1] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 770-778).

[2] Zhang B, Xiong D, Su J. Neural machine translation with deep attention. IEEE transactions on pattern analysis and machine intelligence. 2018 Oct 16;42(1):154-163.

[3] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016 Nov 10.

[4] Cao W, Wang X, Ming Z, Gao J. A review on neural networks with random weights. Neurocomputing. 2018 Jan 31;275:278-87.

[5] Schmidt WF, Kraaijveld MA, Duin RP. Feed forward neural networks with random weights. In International Conference on Pattern Recognition 1992 Aug 30 (pp. 1-4).

[6] Pao YH, Takefuji Y. Functional-link net computing: theory, system architecture, and functionalities. Computer. 1992 May;25(5):76-79.

[7] Huang GB, Zhu QY, Siew CK. Extreme learning machine: a new learning scheme of feedforward neural networks. In2004 IEEE international joint conference on neural networks 2004 Jul 25 (pp. 985-990).

[8] Wang D, Li M. Stochastic configuration networks: Fundamentals and algorithms. IEEE transactions on cybernetics. 2017 Aug 21;47(10):3466-3479.

[9] Chen CP, Liu Z. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. IEEE transactions on neural networks and learning systems. 2017 Jul 21;29(1):10-24.

[10] Huang GB, Chen L, Siew CK. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans. Neural Networks. 2006 Jul 1;17(4):879-892.

[11] Chen CP, Liu Z, Feng S. Universal approximation capability of broad learning system and its structural variations. IEEE transactions on neural networks and learning systems. 2018 Sep 10;30(4):1191-1204.

[12] Cao W, Patwary MJ, Yang P, Wang X, Ming Z. An initial study on the relationship between meta features of dataset and the initialization of nnrw. In2019 International Joint Conference on Neural Networks (IJCNN) 2019 Jul 14 (pp. 1-8). IEEE.

[13] Zhai J, Zhang S, Wang C. The classification of imbalanced large data sets based on mapreduce and ensemble of elm learners. International Journal of Machine Learning and Cybernetics. 2017 Jun 1;8(3):1009-17.

[14] Xu S, Wang J. A fast incremental extreme learning machine algorithm for data streams classification. Expert systems with applications. 2016 Dec 15;65:332-44.

[15] Qiu X, Suganthan PN, Amaratunga GA. Ensemble incremental learning random vector functional link network for short-term electric load forecasting. Knowledge-Based Systems. 2018 Apr 1;145:182-96.

[16] Mirza B, Lin Z, Liu N. Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift. Neurocomputing. 2015 Feb 3;149:316-29.

[17] Wang D, Cui C. Stochastic configuration networks ensemble with heterogeneous features for large-scale data analytics. Information Sciences. 2017 Nov 1;417:55-71.

[18] Cao W, Gao J, Ming Z, Cai S, Zheng H. Impact of probability distribution selection on RVFL performance. In International Conference on Smart Computing and Communication 2017 Dec 10 (pp. 114-124).

[19] Zhang L, Suganthan PN. A comprehensive evaluation of random vector functional link networks. Information sciences. 2016 Nov 1;367:1094-105.

[20] Cao W, Gao J, Ming Z, Cai S. Some tricks in parameter selection for extreme learning machine. InIOP conference series: materials science and engineering 2017 Oct (Vol. 261, No. 1, p. 012002).