Comments on "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.
Email: caoweipeng@szu.edu.cn

Abstract. This paper comments on our recently published conference paper entitled "An Initial Study on the Relationship Between Meta Features of Dataset and the Initialization of NNRW". We point out that the above-mentioned article has a typographical error in proving that using Gamma distribution to initialize NNRW is not a good choice, and give the corresponding correct proof.

Keywords: Randomized neural networks, neural networks with random weights, random vector functional link network, meta features.
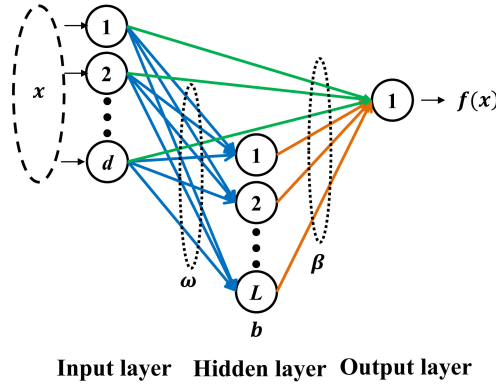
1 Introduction

Neural networks with random weights (NNRW) have received extensive attention in recent years, and relevant algorithms and applications have shown great potential in many fields [3, 5]. Some notable algorithms include: random vector functional link network (RVFL) [8], extreme learning machine (ELM) [7], stochastic configuration network (SCN) [10], broad learning system (BLS) [4], etc.

In traditional neural networks (e.g., BP [9] and CNN [6]), all parameters in the network must be fine-tuned iteratively until the model error converges to a preset threshold. The most commonly used method to achieve this goal is gradient descent and its variants.

Different from the traditional neural networks, some parameters in NNRW remain unchanged during the model training after randomized initialization according to certain rules, and the remaining parameters are obtained analytically. Specifically, taking NNRW with a single-hidden-layer network structure as an example (as shown in Fig. 1 [2]), its input weights w and hidden biases b are assigned randomly from a given range (e.g., [-1, 1]) and kept frozen throughout the training process, while its output weights β are obtained analytically (e.g., using the least squares method).

The training process of NNRW is non-iterative, which can make the model achieve extremely fast training speed and good generalization ability when the size of the training data is relatively small and the data features are relatively

Fig. 1. The network structure of a single-hidden-layer NNRW [2].

simple. Thanks to such advantages, NNRW-related algorithms have been applied in many scenarios [3].

It can be inferred from the above introduction that the feature extraction quality of NNRW depends largely on the effect of its random feature mapping. In other words, the choice of w and b have a significant impact on model performance. In fact, how to choose the appropriate w and b when modeling is one of the most important open issues in this field. Although some researchers have tried to explore and study this issue, there is currently no simple and convenient method to guide the assignment of these random parameters. For example, Wang et al. [10] pointed out that the selection of w and b should depend on the training data and they proposed a supervisory mechanism to guide the assignment of these parameters. Although this method is effective in some specific scenarios, its computational complexity is relatively high, specially when the number of candidate parameters is set to be large.

Similar to Wang's work, we have empirically found that the most suitable random intervals for different data sets are also different [1]. Further, we studied the relationship between meta-features of a data set and the initialization of NNRW in [2]. Both of these works provide valuable guidelines for the initialization of NNRW.

However, in our further study of the above problem, we found that there is a typographical error in the equation (8) in [2], which may be due to negligence during the proofreading process. To avoid misleading readers, we amend this problem in this paper.

2   Claims

In [2], the following equation is used to explain why the Gamma distribution is not an ideal choice for initializing the NNRW (see [[2]] for details).

$$
\begin{aligned}
Var(w_i x + b_i) &= Var(w_i x) + Var(b_i) \\
&= E(w_i x)^2 - E(w_i^2 x^2) + Var(b_i) \\
&= E(w_i)^2 E(x)^2 - E(w_i^2) E(x^2) + Var(b_i) \\
&= E(w_i)^2 E(x)^2 - (Var(w_i) + E(w_i)^2) E(x^2) + Var(b_i) \\
&= Var(x) E(w_i)^2 - Var(w_i) E(x)^2 + Var(b_i).
\end{aligned}
\tag{1}
$$

In fact, the correct version should be as follows:

$$
\begin{aligned}
Var(w_i x + b_i) &= Var(w_i x) + Var(b_i) \\
&= E(w_i^2 x^2) - [E(w_i x)]^2 + Var(b_i) \\
&= E(w_i^2) E(x^2) - [E(w_i)]^2 [E(x)]^2 + Var(b_i) \\
&= (Var(w_i) + [E(w_i)]^2) E(x^2) - [E(w_i)]^2 [E(x)]^2 + Var(b_i) \\
&= Var(w_i) E(x^2) + [E(w_i)]^2 E(x^2) - [E(w_i)]^2 [E(x)]^2 + Var(b_i) \\
&= Var(w_i) E(x^2) + [E(w_i)]^2 (E(x^2) - [E(x)]^2) + Var(b_i) \\
&= Var(w_i) E(x^2) + [E(w_i)]^2 Var(x) + Var(b_i)
\end{aligned}
\tag{2}
$$

3 Conclusions

In this paper, we corrected a typographical error in [2] and briefly reviewed its research motivation. Readers who are interested in the details of related papers can refer to the original ones.

References

1. W. Cao, J. Gao, Z. Ming, and S. Cai, *Some tricks in parameter selection for extreme learning machine*, in IOP conference series: materials science and engineering, vol. 261, IOP Publishing, 2017, p. 012002.
2. W. Cao, M. J. Patwary, P. Yang, X. Wang, and Z. Ming, *An initial study on the relationship between meta features of dataset and the initialization of nnrw*, in 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.
3. W. Cao, X. Wang, Z. Ming, and J. Gao, *A review on neural networks with random weights*, Neurocomputing, 275 (2018), pp. 278–287.
4. C. P. Chen and Z. Liu, *Broad learning system: An effective and efficient incremental learning system without the need for deep architecture*, IEEE transactions on neural networks and learning systems, 29 (2017), pp. 10–24.
5. 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 alzheimers disease*, in Thirty-First AAAI Conference on Artificial Intelligence, 2017.

6. Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, *Filter pruning via geometric median for deep convolutional neural networks acceleration*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4340–4349.

7. G.-B. Huang, L. Chen, C. K. Siew, et al., *Universal approximation using incremental constructive feedforward networks with random hidden nodes*, IEEE Trans. Neural Networks, 17 (2006), pp. 879–892.

8. Y.-H. Pao and Y. Takefuji, *Functional-link net computing: theory, system architecture, and functionalities*, Computer, 25 (1992), pp. 76–79.

9. D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, *Backpropagation: The basic theory*, Backpropagation: Theory, architectures and applications, (1995), pp. 1–34.

10. D. Wang and M. Li, *Stochastic configuration networks: Fundamentals and algorithms*, IEEE transactions on cybernetics, 47 (2017), pp. 3466–3479.