http://math-sci.ui.ac.ir

Journal of Mathematics and Society
ISSN (print): 2345-6493, ISSN (on-line): 2345-6507
Vol. 10 No. 4 (2025), pp. 1–24.

http://www.ui.ac.ir

BAYESIAN NEURAL NETWORKS; WHY AND HOW?

MOEIN MONEMI, S. MAHMOUD TAHERI, S. MORTEZA AMINI*

Abstract. One of the main challenges in utilizing neural networks is the problem of overfitting. This occurs when a neural network model fits the training data too precisely, but fails to generalize to data outside the training set. This lack of generalization is often observed when the number of training samples is smaller than the number of features being analyzed, and the complexity of the model — that is, the number of weights and biases in the neural network — is high. In such situations, ensemble learning, and more specifically bagging methods, are commonly employed. These methods use resampling techniques to incorporate uncertainty into the model, thereby improving the model's ability to generalize. However, when the training sample size is extremely limited, resampling becomes less effective, and the uncertainty introduced in the model is very limited. Bayesian neural networks address this by quantifying parameter uncertainty and considering parameter states that may not have been observed in the existing data. This leads to a significant improvement in model generalization. In addition to mitigating overfitting, this approach also provides access to the posterior predictive distribution, allowing for the calculation of prediction intervals. In this article, we briefly review Bayesian neural networks, explain how they are trained, and then analyze data and compare these models with standard neural networks.

Keywords: Artificial neural networks, Bayesian inference, Posterior distribution, MCMC, Multiple linear regression, Classification

Article Type: Promotional Paper.

Communicated by Afshin Parvardeh.

Received: 03-06-2024, Accepted: 14-10-2024, Published Online: 29-06-2025.

Cite this article: M. Monemi, S. M. Taheri, S. M. Amini, Bayesian Neural Networks; why and how?, *Journal of Mathematics and Society*, **10** no. 4 (2025) 1–24.

http://dx.doi.org/10.22108/msci.2024.141722.1668 .

1. Introduction

Bayesian Neural Networks (BNNs) are extensively used in statistical machine learning due to their ability to use prior knowledge about network parameters, such as weights and biases. One of the main challenges in traditional neural networks is the problem of overfitting, which occurs when the model fits the training data too precisely, thereby failing to generalize well to unseen test data. BNNs, by quantifying uncertainty, allow for consideration of parameter states that may not have been seen in the training data, thereby improving model robustness. In addition to reducing overfitting, BNNs employ the posterior predictive distribution to predict new data and derive prediction intervals, which offer a measure of uncertainty in the predictions. Nevertheless, obtaining the posterior distribution poses significant challenges due to the intractable integrals in the denominator of Bayes' theorem—an issue particularly pertinent in models like BNNs, which rely on activation functions, as illustrated in Figures 2 and 3. To overcome this, approximate inference methods such as Markov Chain Monte Carlo (MCMC), Variational Bayes, and Expectation Propagation are utilized to approximate the posterior distribution.

In this paper, we review Bayesian Neural Networks that utilize a multilayer perceptron, similar to those depicted in Figures 1 and 4, and detail the process of training them using the MCMC method, specifically employing the Metropolis-Hastings algorithm [8]. This method generates a sequence of dependent random samples, that typically exhibit a first-order Markov property. In first-order Markov processes, the conditional distribution of any state, given all previous states, depends only on the previous state. One of the most well-known methods for achieving this, is the Metropolis-Hastings algorithm introduced by [8], which uses a proposal distribution to generate samples from the unknown distribution such as the posterior distribution, which we will implement for both regression and classification tasks.

In the regression task, we assume we have the inputs variable $X \in \mathbb{R}^{n\times d}$ and the outputs variable $y \in \mathbb{R}^{n\times p}$, where n is the number of inputs, and d and p represent the dimensionality of the input and output variables, respectively. Our goal is to model $\hat{y} = f(X) + \epsilon$, with $\epsilon \sim \mathcal{N}_n(0, \sigma^2 I_n)$, requiring us to approximate the posterior distribution $p(w, b, \sigma^2|X, y)$. Here, w and b represent the weights and biases of the neural network, and σ denotes the standard deviation of the likelihood function $p(y|X, W, b, \sigma^2)$, where $y|X, W, b, \sigma^2 \sim \mathcal{N}_n(f(X), \sigma^2 I_n)$. The Metropolis-Hastings algorithm in this task uses proposal distributions for the posterior parameters (w, b, σ^2) and generates samples accordingly. After T iterations, a density estimator is used to approximate the posterior distribution.

In the classification task, similar to regression, we have input variables $X \in \mathbb{R}^{n\times d}$ and output variables $y \in \mathbb{R}^{n\times k}$, where k represents the number of classes. Each row of y is a basis vector, with only one element equal to 1 and the rest equal to 0. The objective in this problem is to classify

http://dx.doi.org/10.22108/msci.2024.141722.1668

the inputs X based on the given data. For this task, we aim to estimate the posterior distribution $p(w, b|X, y)$.

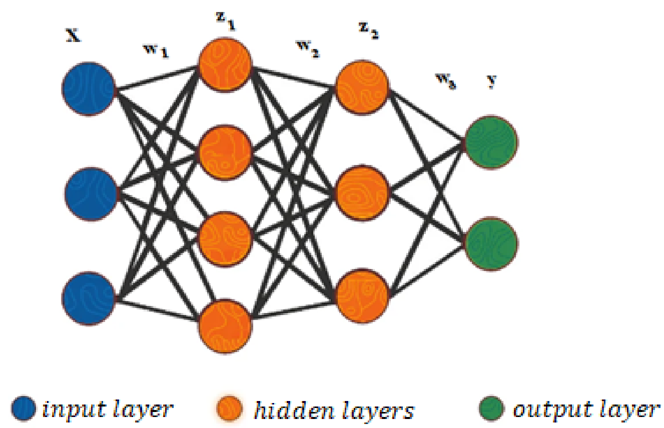

FIGURE 1. The architecture of two hidden layers Neural Network

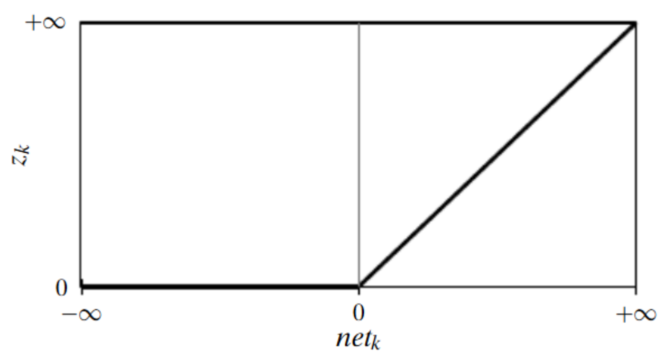

FIGURE 2. Relu activation function [18]

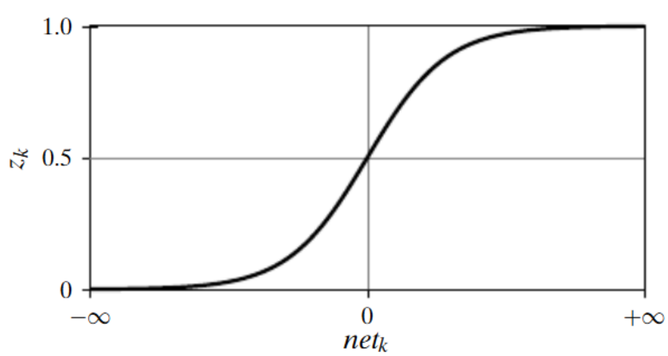

FIGURE 3. Sigmoid activation function [18]

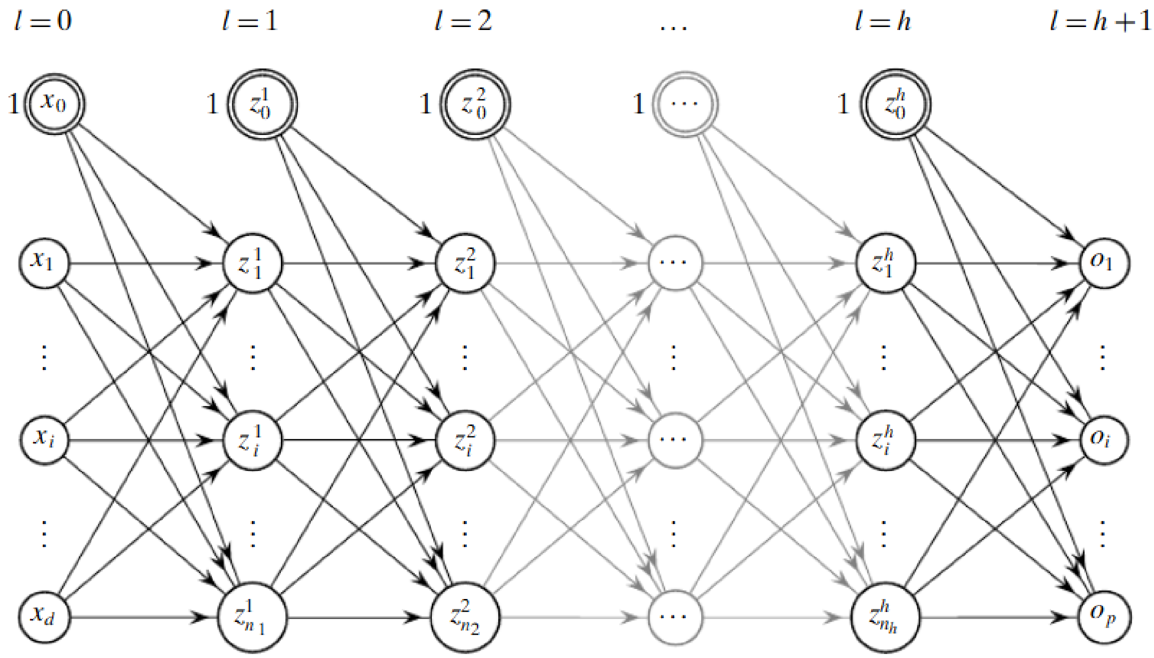

FIGURE 4. The architecture of the Neural Network consists of $h+2$ layers, where h represents the number of hidden layers, with the input and output layers completing the structure [18]

2. Main Results

In this section, we evaluate the performance of BNNs against standard neural networks on several datasets. The results demonstrate that BNNs outperform traditional neural networks in both regression and classification tasks. Figures 5 and 6 respectively illustrate the residuals from the regression task and the ROC curve from the classification task. Tables 1 and 2 compare the performance of the two approaches across various evaluation metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), accuracy, F-measure, and others.

	BNN	**ANN**
Mean Square Error	1.21	30.75
Mean Absolute Error	0.88	4.80

TABLE 1. Comparison of the performance of BNN and standard NN on the Riboflavin dataset.

http://dx.doi.org/10.22108/msci.2024.141722.1668

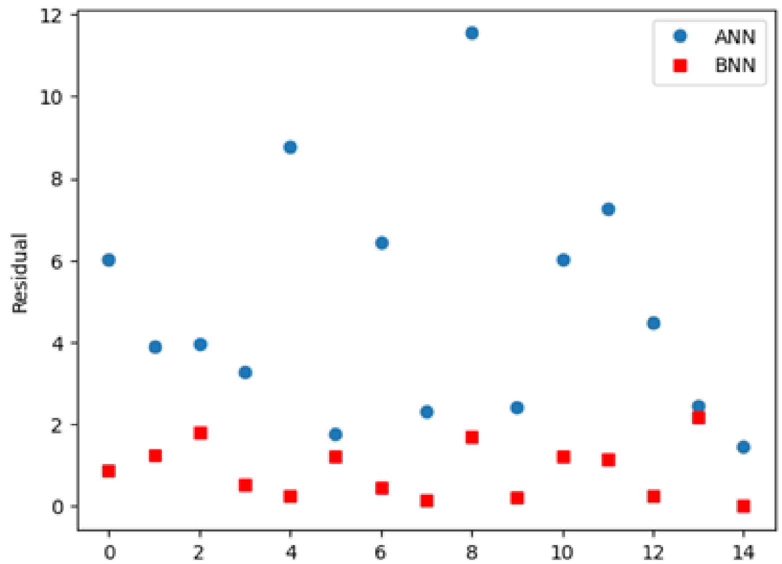

FIGURE 5. Residual values for 15 data points in the regression task on the Riboflavin dataset, comparing BNN (represented by square markers) and standard NN (represented by circle markers)

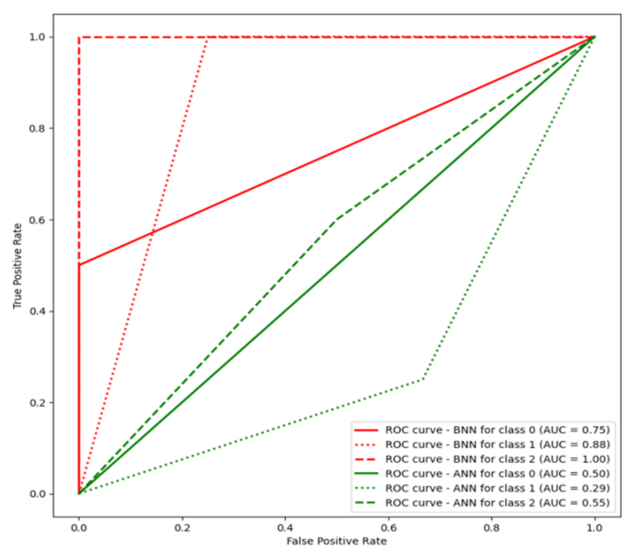

FIGURE 6. ROC curve of BNN versus standard NN in classification task on lung cancer dataset (The continuous curve: for class 0, dotted curve: for class 1, dashed curve: for class 2)

	BNN	**ANN**
Accuracy	0.85	0.42
F-measure	0.84	0.38
F-Beta measure	0.86	0.36
Jaccard measure	0.75	0.27

Table 2. Evaluation of BNN and NN on the lung cancer dataset based on four common metrics.

3. Conclusions

Bayesian neural networks was reviewed as an efficient alternative to classical neural networks. In two practical examples, it was observed that under certain conditions, Bayesian neural networks perform significantly better than classical neural networks. Since there is no conjugate prior and posterior distribution for neural networks based on the activation functions used, the Metropolis-Hastings sampling method was employed to approximate the posterior distribution. It should be noted that, this method incurs high computational costs when dealing with high-dimensional parameters. Therefore, recent research has shifted toward using more cost-effective posterior approximation methods, such as Variational Bayes, [1], [12], [16], and [18]. Future work could explore faster approximation methods (such as Variational Bayes) and improvements in sampling methods like $MCMC$ for Bayesian neural networks.

References

[1] C. Blundell, J. Cornebise, K. Kavukcuoglu and D. Wierstra, Weight uncertainty in neural network, *Proc. of the International Conference on Machine Learning PMLR*, Lille, France, (2015) 1613–1622.

[2] J. P. Bharadiya, A review of Bayesian machine learning principles, methods, and applications, *Int. J. Innov. Sci. Res. Technol.*, **8** no. 5 (2023) 2033–2038.

[3] R. Chandra, R. Chen and J. Simmons, Bayesian neural networks via MCMC: a Python-based tutorial, (2023). https://doi.org/10.48550/arXiv.2304.02595.

[4] C. M. Carlo, Markov chain monte carlo and gibbs sampling, *Lecture Notes for EEB 581*, (2004) 24 p.

[5] A. Graves, Practical variational inference for neural networks, Part of *Part of Advances in Neural Information Processing Systems 24 (NIPS)*, (2011)

[6] A. Gelman, J. B. Carlin, H. S. Stern and D. B. Rubin, *Bayesian Data Analysis*, Chapman and Hall/CRC, 1995.

[7] Z. Q. Hong and J. Y. Yang, Lung cancer, UCI Machine Learning Repository, (1992). https://doi.org/10.24432/C57596.

[8] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, **57** no. 1 (1970) 97–109.

[9] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine and M. Bennamoun, Hands-on Bayesian neural networks—A tutorial for deep learning users, *IEEE Computational Intelligence Magazine*, **17** no. 2 (2022) 29–48.

[10] H. D. Kabir, A. Khosravi, M. A. Hosen and S. Nahavandi, Neural network-based uncertainty quantification: A survey of methodologies and applications, *IEEE Access*, **6** (2018) 36218–36234.

[11] J. Ker, L. Wang, J. Rao and T. Lim, Deep learning applications in medical image analysis, *IEEE Access*, **6** (2018) 9375–9389.

[12] I. Oleksiienko, D. T. Tran and A. Iosifidis, Variational neural networks, *Procedia Computer Science*, **222** (2023) 104–113.

[13] M. R. Meshkani and A. Kavousi Dolanghar, Bayesian Statistical Methods, *Shahid Beheshti University of Medical Sciences Press*, (2022). [In Persian]

[14] C. P. Robert, G. Casella and G. Casella, *Monte carlo statistical methods*, **2**, Springer, 1999.

[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus, Intriguing properties of neural networks, (2013). arXiv preprint arXiv:1312.6199. https://doi.org/10.48550/arXiv.1312.6199

[16] S. Sun, G. Zhang, J. Shi and R. Grosse, Functional variational Bayesian neural networks, (2019). arXiv preprint arXiv:1903.05779. https://doi.org/10.48550/arXiv.1903.05779.

[17] S. M. Taheri, Statistics and artificial neural networks, *In Proceedings of the 8th Iranian Statistics Conference*, Shiraz University, (2006) 81–91. [In Persian]

[18] M. N. Tran, T. N. Nguyen, and V. H. Dao, A practical tutorial on variational Bayes, (2021) 43 p. arXiv preprint arXiv:2103.01327. https://doi.org/10.48550/arXiv.2103.01327

[19] M. J. Zaki and W. Meira Jr, *Data mining and machine learning: fundamental concepts and algorithms*, Cambridge University Press, 2020.

Moein Monemi

School of Engineering Science, College of Engineering, University of Tehran Tehran, Iran

Email: moein.monemi@ut.ac.ir

S. Mahmoud Taheri

School of Engineering Science, College of Engineering, University of Tehran Tehran, Iran

Email: sm_taheri@ut.ac.ir

S. Morteza Amini

Department of Statistics, School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran Tehran, Iran

Email: morteza.amini@ut.ac.ir