Advances in Computer, Signals and Systems (2022)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2022.060612
ISSN 2371-8838 Vol. 6 Num. 6

A Survey on Bayesian Neural Networks

Hua Zhong[1,a], Lin Liu[1,b,*], Shicheng Liao[1,c]

[1]*Institute of Problem Solving, Yunnan Normal University, Kunming, China*
[a]*zzzhhh0813@163.com*, [b]*liulinrachel@163.com* ,[c]*csscliao@163.com*
[*]*Corresponding author*

Keywords: bayesian, neural network, variational inference, MCMC

Abstract: Bayesian Neural Network is the product of combining Bayesian and neural networks, which is one of the most popular neural network branches in deep learning and is widely used in model building in various fields. Bayesian inference is mainly divided into two types: variational inference and MCMC methods. Variational inference uses a pre-defined simple distribution to approximate the posterior distribution, which is faster and suitable for larger scale data. MCMC approach approaches the posterior distribution by sampling, which is more accurate, but relatively slower. This paper introduces three more representative Bayesian models and briefly introduces the main steps and formulas involved in the models. In addition, a variety of Bayesian neural network models are categorized and organized. Finally, the development of Bayesian is summarized and prospected.

1. Introduction

Neural network is a kind of network that simulates the mechanism of human brain to model. Human research on neural network has been long, and with the deeper research on neural network, deep neural network (DNN), generative adversarial network (GAN) and convolutional neural network (CNN) have gradually appeared. The training results of neural networks in the image domain show that overfitting due to training can cause the model to achieve very good classification results on training data, but unsatisfactory results on prediction data. This is because the model makes an overconfident judgment. Therefore, in order to solve the uncertainty problem of training data, many researches to fuse Bayesian framework with neural networks are pulled off.

2. Current Status of Domestic and International Research

There are two main methods used to do Bayesian inference: Markov Chain Monte Carlo (MCMC) based on sampling and Approximate Variational Inference (VI) method. In MCMC method, No need to assuming the distribution of the model, the posterior distribution is approximated by random sampling in the probability space. In contrast, the Variational Inference method sets a simple distribution in advance and makes the simple distribution close to the posterior distribution by minimizing the KL dispersion between the simple distribution and the posterior distribution. The MCMC method does not need to assume the model in advance compared with the

Variational Inference, and although it obtains more accurate results, it has the problems of high variance and longer time. Variational inference is faster in computation, but less accurate compared with MCMC, and more suitable for larger scale statistical problems. In this section, an example study of two Bayesian inference methods is briefly presented.

The first is the study of Bayesian inference using MCMC method.In 2017, Baele G [1] et al. in order to reduce the computational burden and solve the problem of large number of parameters estimation, they proposed a Markov chain Monte Carlo (MCMC) method using adaptive multivariate transition kernel to estimate a large number of parameters.The model performance is improved by 14 times, which is very significant, but the hardware support is needed to improve the computational speed, and storage and maintenance will be one of the main limitations of this approach. In 2017, Saatci Y [2] et al. proposed a model called BGAN (Bayesian generative adversarial network), which uses stochastic gradient Hamiltonian Monte Carlo to marginalize the weights of the generator and discriminator, effectively avoids pattern collapse of GAN. Generated samples become diverse and interpretable, and the semi-supervised learning reduces the dependence of deep learning on labeled data. In 2019, Wang et al [3] proposed a model combining artificial neural network (ANN) and MCMC called ANN-MCMC, which significantly reduces the computational cost of the algorithm. Since the traditional Markov chain has a high sample rejection rate when computing the posterior distribution of insensitive parameters, while the convergent ANN can generate a large number of low-cost samples, the model is suitable for computational systems with limited computational power.

Second, research on models based on variational inference is presented. An example is the BNN proposed by Bittig H C [19] and others, applied to tasks such as picture classification. The method eliminates the use of dropout, incorporates uncertainty and normalized measures, and predictions are built on cognitive uncertainty and stochastic uncertainty. It effectively reduces the uncertainty in the neural network and reduces the computational and time costs. The authors state that reducing the number of model parameters enables better generalization of Bayes. This article appears laid the foundation for subsequent research on Bayesian convolutional neural networks, Bayesian generative adversarial networks, Bayesian recurrent neural networks, etc. Immediately in 2019, Chien J T [5] et al. gave research results related to variational Bayesian inference and generative adversarial networks. To address the problem of GAN pattern collapse and to compensate for the error, the variational estimation of the generator and discriminator enables the model to achieve superior results. Their experimental results show that the Bayesian variational inferential generative adversarial network (VGAN) model has good performance on both unsupervised and semi-supervised learning. In summary, it can be seen that Bayesian framework is an effective uncertainty inference method.

3. Three Principles of Combining Bayesian and Neural Networks

This section will introduce three principles of combining Bayesian and neural networks, sort out the main formulas involved in the implementation of the models, and describe the main features of the models. The three models include BNN and BCNN using variational inference, BGAN using MCMC.

3.1. Bayesian Neural Network (BNN)

BNN is a kind of neural network used to optimize the weight parameters. Unlike traditional neural networks, BNN give not only the prediction results but also the uncertainty of the prediction results. In BNN, the neural network weight parameters are treated as random variables, and its

construction is shown in Figure 1. Adding a priori is equivalent to adding a constraint and a canonical to the network, which effectively avoids the overfitting and overconfidence problems of traditional neural networks. The key parameters of the network in BNN are described by a priori, such as $p(w)$. Denoting the likelihood function by $p(D|w)$, the posterior is $p(w|D)$. Where W is the weight and D is the observed data, the output of the integrated network is synthetically expressed as Equation (1).

$$p(w|D) = \frac{p(w)p(D|w)}{p(D)} \tag{1}$$

where $p(D)$ denotes the edge likelihood, and from equation (1) it can be seen that the key to BNN modeling is to do approximate the posterior inference, if the direct sampling of the posterior probability exists the posterior distribution multidimensional, so the variational inference is used to approximate the posterior distribution with a simple distribution. Consider each weight w_i as random and w_i is sampled from a normal distribution $\theta \sim N(\mu_i, \sigma_i)$. A simple distribution $q(w|\theta)$ is used to approximate the posterior distribution, at which point the KL scatter is used to measure the distance between $q(w|\theta)$ and $p(D|w)$ as follows.

$$\theta^* = \arg\min_{\theta} KL[q(w|\theta) \| p(w|D)] \tag{2}$$

Immediately afterwards, the reparameterization trick is used to sample w. In addition, the value of \theta has to be guaranteed, so σ has to be sampled, and finally $w \sim N(\mu, \log(1+e^{\rho}))$ is obtained.

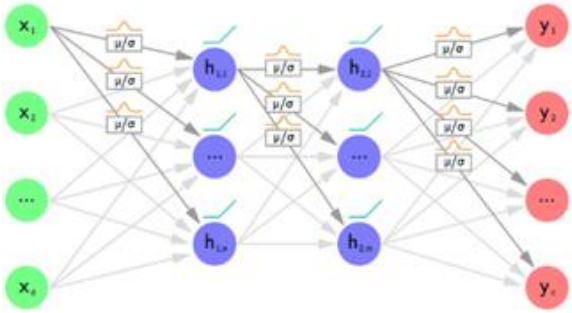

Figure 1: BNN (this figure is from the work of Shridhar K [4] et al)

BNN is to approximate the posterior distribution by variational inference, which turns the W in traditional neural networks into a probability distribution with very strong robustness. However, BNN is not effective in the face of large-scale data, and due to the decreasing importance of the prior in large-scale data, the effect of BNN will be essentially the same as that of traditional neural networks.

3.2. Bayesian Convolutional Neural Network (BCNN)

This section introduces the framework structure of Bayesian convolutional neural network, referring to [7], the principle of BCNN is basically the same as BNN. Therefore, the same probability distribution is used in BCNN to describe the weights of the network, but unlike the normal neural network, the convolutional neural network requires convolutional operations. So it is also necessary to represent the weights on the convolutional layer with probability distribution. As shown in Figure 2, it can be seen that the left side is a traditional CNN with a single value of the convolution kernel, while the right side is a BCNN, which represents the single value of the

convolution kernel in a CNN as a probability distribution. Sampling from the distribution is done when forward propagation is performed, but this goes back to what was said above about direct sampling being very difficult, so here too it is necessary to use reparameterization trick.

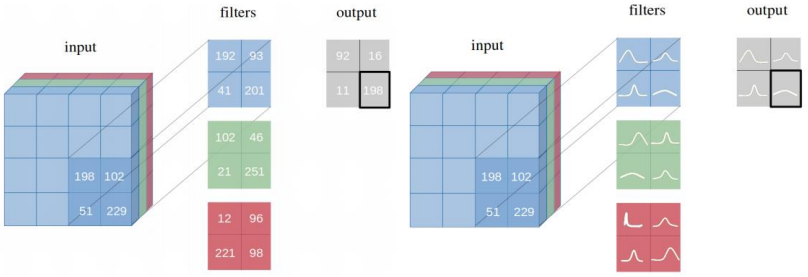

Figure 2: CNN and BCNN (this figure is from the work of Shridhar K [4] et al)

Thus the sampling of the weights can be expressed as $f(\varepsilon) = w = \mu + \varepsilon \times \sigma, \varepsilon \in N(0,1)$, which is equivalent to sampling from $N(\mu, \sigma^2)$. Similarly, BCNN uses a simple distribution to approximate the posterior distribution, and the process is expressed as Equation (3).

$$p_D(y^* \mid x^*) = \int p_w(y^* \mid x^*) p_D(w) dw = \int Cat(y^* \mid f_w(x^*)) N(w \mid \mu, \sigma^2) dw$$

$$E_q[p_D(y^* \mid x^*)] = \int q_\theta(w \mid D) p_w(y \mid x) dw \approx \frac{1}{T} \sum_{t=1}^{T} p_{w_t}(y^* \mid x^*) \quad (3)$$

The number of Monte-Carlo samples is T. According to the above reasoning, BCNN can be implemented by using the reparameterization trick, which adds Bayesian to CNN, which will be more consistent with the human brain's way of thinking and will make the results more accurate.

3.3. Bayesian Generative Adversarial Networks

Generative adversarial networks are an excellent unsupervised and semi-supervised learning framework, and the combination with Bayesian also produces a wonderful chemistry.In 2017, Chien J T [5] et al. proposed BGAN to model using conditional posterior distributions, using dynamic gradient Hamiltonian Monte Carlo method to maximize the weights in generative and discriminative networks, effectively avoiding model collapse and providing excellent semi-supervised learning quantitative results. For example, Figure 3 shows the samples generated by setting parameters for two different styles of handwritten digits, and it can be seen that BGAN is able to retain the full probability distribution over the parameters, however, the full probability distribution represented by the traditional GAN using point estimates tends to lose potentially valuable data.

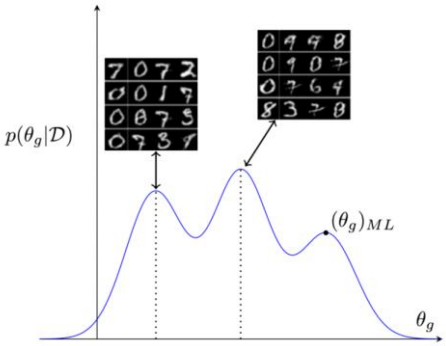

Figure 3: This figure is from the work of Saatci Y [2] et al.

BGAN [5] sets distributions on the generative and discriminative network parameters θ_g and θ_d to obtain $p(\theta_g \mid \alpha_g)$ and $p(\theta_d \mid \alpha_d)$, sampling the two parameters from the distributions, the posterior inference of θ_g and θ_g under unsupervised learning is expressed as Equation (4) and Equation (5).

$$p(\theta_g \mid z, \theta_d) \propto (\prod_{i=1}^{n_g} D(G(z^{(i)}; \theta_g); \theta_d)) p(\theta_g \mid \alpha_g) \quad (4)$$

$$p(\theta_d \mid z, x, \theta_g) \propto \prod_{i=1}^{n_d} D(x^{(i)}; \theta_d) \times \prod_{i=1}^{n_g} (1 - D(G(z^{(i)}; \theta_g); \theta_d)) \times p(\theta_d \mid \alpha_d) \quad (5)$$

where α_g and α_d are hyperparameters and n_g and n_d denote the number of samples under one batchsize of the generator and discriminator. Stochastic gradient Hamiltonian Monte Carlo sampling (SGHMC [6]) is used in BGAN because SGHMC is closely related to the momentum-based SGD, which is effective for GAN training and can be implemented by importing SGD parameter settings into SGHMC.

In addition, BGAN extends to semi-supervised learning, where the emergence of BGAN reduces the reliance on labeled data. From the experimental results, BGAN can effectively capture the data distribution, and the combination of Bayesian and generative adversarial networks has great research significance and research value.

Besides the several models presented above, the combined products of Bayesian and neural networks are used in various research areas. For example, Bayesian RNNs [8] for language description and image description, Bayesian Graph Neural Networks for multi-instance learning, meta-learning[9-10], Bayesian Graph Convolutional Neural Networks [11] applied to the recommendation domain, etc. We have classified some models as shown in Table 1.

Table 1: Bayesian inference approach of the models

Models	MCMC	VI
Bayesian gan [2]	√	
Variational bayesian gan [5]		√
An application to analyzing partitioned data in BEAST[1]	√	
Bayesian convolutional neural network [4]		√
Bayesian neural network [19]		√
Bayesian recurrent neural networks[8]		√
Multiple Instance Learning using Bayesian Graph Neural Networks[9]		√
Continual Meta-Learning with Bayesian Graph Neural Networks[10]		√
A Framework for Recommending Accurate and Diverse Items Using Bayesian Graph Convolutional Neural Networks[11]	√	
Deep Latent Dirichlet Allocation[12]	√	
Dirichlet Belief Networks[13]	√	
Gaussian Mixture Variational Autoencoder[14]		√
Attend, Infer, Repeat [15]		√
Deep Variational Bayes Filters [16]		√
DeepAR [17]		√
Embed to Control [18]		√

4. Conclusion

With the development of information technology and the update of hardware devices, neural networks have come back into the public view as a research hotspot, but most of the neural network models are black-box models, which lack interpretability. Although neural networks have excellent learning ability, they also have problems such as overfitting. In order to solve the problems of model overfitting and poor interpretability, Bayesian has become one of the key concerns in uncertainty research, and the combination of Bayesian and neural networks has brought a new light to the research. According to the research, Bayesian is mainly used to achieve accurate prediction by placing probability distributions on the weights of neural networks, using simple distributions to approximate complex posterior distributions or using MCMC method.

At present, the more common combination of Bayesian and neural networks are BNN, BCNN and BGAN, whose common features are to do research from the weights of neural networks. With the deepening of researchers' research on Bayesian and neural networks, many Bayesian neural network frameworks have emerged, such as pyro, pymc, Blitz, bead, BoTorch, Edward, and Tensorflow Probability. the emergence of these frameworks has greatly facilitated our use of Bayesian neural networks, and applied Bayesian neural networks to more domains, especially in tasks with small sample sizes, where model performance can be better improved. Accordingly, the problems of overfitting and poor model interpretability can be solved. Overall, Bayesian uncertainty inference process is more in line with the human brain's judgment of things, while the neural network has excellent learning ability, the combination of the two makes up for the shortcomings of traditional neural networks, opening up a new path for research, and it is believed that Bayesian neural networks will shine in more fields in the future.

References

[1] Baele G, Lemey P, Rambaut A, Suchard MA. Adaptive MCMC in Bayesian phylogenetics: an application to analyzing partitioned data in BEAST. Bioinformatics. 2017 Jun 15;33(12):1798-1805. doi: 10. 1093 /bioinformatics/ btx088. PMID: 28200071; PMCID: PMC6044345.

[2] Saatci Y, Wilson A G. Bayesian gan[J]. Advances in neural information processing systems, 2017, 30.

[3] Wang, Jiaxing & Zhou, Zijun & Lin, Keli & Law, Chung & Yang, Bin. (2020). Facilitating Bayesian analysis of combustion kinetic models with artificial neural network. Combustion and Flame. 213. 87-97. 10. 1016/ j .combustflame.2019.11.035.

[4] Shridhar K, Laumann F, Liwicki M. A comprehensive guide to bayesian convolutional neural network with variational inference[J]. arXiv preprint arXiv:1901.02731, 2019.

[5] Chien J T, Kuo C L. Variational bayesian gan[C]//2019 27th European Signal Processing Conference (EUSIPCO). IEEE, 2019: 1-5.

[6] Chen T , Fox E B , Guestrin C . Stochastic Gradient Hamiltonian Monte Carlo[J]. Eprint Arxiv, 2014:1683-1691.

[7] Liu Ze-rui. Tea classification method based on Bayesian convolutional neural network [D]. Tianjin University of Commerce.

[8] Fortunato M , Blundell C , Vinyals O . Bayesian Recurrent Neural Networks[J]. 2017.

[9] Pal S , Valkanas A , Regol F , et al. Bag Graph: Multiple Instance Learning using Bayesian Graph Neural Networks[J]. 2022.

[10] Luo Y , Huang Z , Zhang Z , et al. Learning from the Past: Continual Meta-Learning with Bayesian Graph Neural Networks[C]// National Conference on Artificial Intelligence. Association for the Advancement of Artificial Intelligence (AAAI), 2020.

[11] Sun J , Guo W , Zhang D , et al. A Framework for Recommending Accurate and Diverse Items Using Bayesian Graph Convolutional Neural Networks[C]// KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, 2020.

[12] Cong Y . Deep Latent Dirichlet Allocation with Topic-Layer-Adaptive Stochastic Gradient Riemannian MCMC:, 10.48550/arXiv.1706.01724[P]. 2017.

[13] Fan X , Li B , Li Y , et al. Poisson-Randomised DirBN: Large Mutation is Needed in Dirichlet Belief Networks[C]// International Conference on Machine Learning. PMLR, 2021.

[14] Zhou C , Ban H , Zhang J , et al. Gaussian Mixture Variational Autoencoder for Semi-Supervised Topic Modeling[J]. IEEE Access, 2020, PP(99):1-1.
[15] Eslami S , Heess N , Weber T , et al. Attend, Infer, Repeat: Fast Scene Understanding with Generative Models:, 10.48550/arXiv.1603.08575[P]. 2016.
[16] Karl M , Soelch M , Bayer J , et al. Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data[J]. 2017.
[17] Flunkert V , Salinas D , Gasthaus J . DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks:, 10.1016/j.ijforecast.2019.07.001[P]. 2020.
[18] Watter M , Springenberg J T , Boedecker J , et al. Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images[J]. Advances in neural information processing systems, 2015.
[19] Bittig H C , Steinhoff T , Claustre H , et al. Bayesian Neural Networks[J]. Frontiers in Marine ence, 2018, 5.