Bayesian Convolutional Neural Networks with Variational Inference

Yi-Pei Chan
Department of Statistics
Columbia University
New York, NY 10027
yc3700@columbia.edu

Abstract

In this report, we investigated recent works in using Bayesian Convolutional Neural Networks with variational inference to perform image classification tasks. Based on ideas introduced by Shridhar et al. (1), we reproduced the Bayesian LeNet model and tested the performance of our network on image classification tasks with MNIST, CIFAR-10, and CIFAR-100 datasets. We as well compared the validation accuracies generated with Bayesian approach to those generated with frequentist inference under the LeNet architecture. At the end of the paper, we discussed possible reasons for the discrepancies and provided promising future works that can potentially improve the model performance in the original paper.

1 Introduction

In the field of image classifications, convolutional neural networks (CNNs) have achieved a series of breakthroughs in recent years, even surpassing human-level accuracy in tasks such as the ImageNet Challenge (2; 3; 4). However, CNNs require a large amount of data for regularisation and fail to express uncertainty which would result in overconfident decisions on small datasets or in regions with little or no data (1; 5; 6). In reality, labeled data may be hard and expensive to collect, and in some applications there is no large amount of data readily available. By introducing probability distribution over the network parameters, Bayesian CNNs can easily learn from small datasets, offer uncertainty estimates, and give a regularization effect to the network, which makes the network robust to overfitting (1; 5; 6).

The concept and attempt in incorporating Bayesian methods into neural networks was first studied in 1990s (7), but its theoretical framework and application were successfully implemented in 2015 by Gal and Ghahramani (8). Until now, researchers have developed numerous approximate inference procedures for Bayesian deep learning, including Laplace approximation (9), MC dropout (5), and variational inference (10; 11; 12), whereby the authors in (1) proposed the novel Bayesian CNN model based on a variational inference method called *Bayes by Backprop* (12).

As there are many great works and researches on Bayesian CNNs, we would review past influential literature in the next section. We selected two major topics - Bayes by Backprop, and Local Reparameterization Trick- that were most relevant to the development of the original paper. The concepts and ideas from the previous works were used in building our final model to evaluate results. The rest of the paper is organized as follows. In section 3 we will describe the MNIST, CIFAR-10, and CIFAR-100 datasets, and the framework we used for our implementations. In Section 4 we will discuss results and show how does our Bayesian CNN compared to its equivalent frequentist CNN. Last, our conclusion will point out possible reasons for discrepancies between our results and those of the authors of the original paper, and we would propose few considerations that must be accounted when employing Bayesian CNN.

2 Related Work

2.1 Bayes by Backprop

Bayes by Backprop is a variational inference method introduced by Blundell et al. for learning a probability distribution on the parameters in feedforward neural networks (12). Fortunato et al. further applied the algorithm in recurrent neural networks (13). Given a set of training samples $\mathcal{D} = (x_i, y_i)_i$, consider a neural network as a probabilistic model $P(y|x, w)$, where w is the set of parameters or weights, Bayesian inference for neural networks aims to calculate the posterior distribution $P(w|\mathcal{D})$, and we expect to answer predictive queries when given test data $\hat{\mathbf{x}}$, which is

$$P(\hat{\mathbf{y}} \mid \hat{\mathbf{x}}) = \mathbb{E}_{P(\mathbf{w}|\mathcal{D})}[P(\hat{\mathbf{y}} \mid \hat{\mathbf{x}}, \mathbf{w})] \tag{1}$$

However, the true posterior is intractable, thus an parameter θ of an approximation distribution on the weights $q_\theta(w|\mathcal{D})$ is defined. The variational learning seeks to find the θ that makes the approximation distribution as close to the true posterior as possible, where the closeness can be measured with Kullback-Leibler (KL) divergence.

$$\mathrm{KL}\left[q_\theta(w|\mathcal{D}) \| p(w|\mathcal{D})\right] = \int q_\theta(w|\mathcal{D}) \log \frac{q_\theta(w|\mathcal{D})}{p(w|\mathcal{D})} dw \tag{2}$$

$$\begin{aligned} \Rightarrow \theta^{opt} &= \arg\min_\theta \mathrm{KL}\left[q_\theta(w|\mathcal{D}) \| p(w|\mathcal{D})\right] \\ &= \arg\min_\theta \int q_\theta(w|\mathcal{D}) \log \frac{q_\theta(w|\mathcal{D})}{p(w)p(\mathcal{D}|w)} dw \\ &= \arg\min_\theta \mathrm{KL}\left[q_\theta(w|\mathcal{D}) \| p(w)\right] \\ &\quad - \mathbb{E}_{q(w|\theta)}[\log p(\mathcal{D}|w)] + \log p(\mathcal{D}) \end{aligned} \tag{3}$$

The $\log p(\mathcal{D})$ term is constant in the optimization process. Consequently, the resulting cost function, $\mathcal{F}(\mathcal{D}, \theta)$, also known as variational free energy (14; 15; 16) or the expected lower bound (17; 14; 18), can be break into two terms: $\mathrm{KL}\left[q_\theta(w|\mathcal{D}) \| p(w)\right]$ the complexity cost, and $\mathbb{E}_{q(w|\theta)}[\log p(\mathcal{D}|w)]$ the likelihood cost. The cost function can as well be reinterpreted as a minimum description length loss function in information theoretic prospect (10; 11).

$$\mathcal{F}(\mathcal{D}, \theta) = \mathrm{KL}\left[q_\theta(w|\mathcal{D}) \| p(w)\right] - \mathbb{E}_{q(w|\theta)}[\log p(\mathcal{D}|w)] \tag{4}$$

With stochastic variational method (11; 12), we sample $w^{(i)}$, the i^{th} Monte Carlo sample draw, from the variational posterior $q_\theta(w|\mathcal{D})$ and approximate the exact cost in equation (4) with

$$\mathcal{F}(\mathcal{D}, \theta) \approx \sum_{i=1}^{n} \log q_\theta(w^{(i)}|\mathcal{D}) - \log p(w^{(i)}) - \log p(\mathcal{D}|w^{(i)}) \tag{5}$$

where n is the number of draws.

2.2 Local Reparameterization Trick

Local reparameterization trick developed by Kingma et. al (19) was utilised by the authors in (1) for Bayesian CNNs. When we translate global uncertainty about parameters into a form of local uncertainty which is independent across examples, such type of reparameterization process is called local reparameterization trick. Assume the variational posterior probability distribution

$$q_\theta(w_{ijhw}|\mathcal{D}) = \mathcal{N}(\mu_{ijhw}, \alpha_{ijhw}\mu_{ijhw}^2) \tag{6}$$

where i is the input layer, j is the output layers, h and w are the height and width of any given filter respectively. For a factorized Gaussian posterior on the weights, the posterior for the activations, b, is also factorized Gaussian (19). The authors in (1) followed (19) in sampling layer activation b, the

implied Gaussian distribution directly, rather than sampling the Gaussian weights and compute the resulting activations. The convolutional layer activations b is thus

$$b_j = A_i * \mu_i + \epsilon_j \odot \sqrt{A_i^2 * (\alpha_i \odot \mu_i^2)}\,,\ \epsilon_j \sim \mathcal{N}(0,1) \tag{7}$$

where A_i is the receptive field, $*$ is the convolutional operation, and $\odot$ is the common component-wise multiplication typically used in CNNs. With this local reparameterization trick, the algorithm can yield a gradient estimator more computationally efficient, also leading to an estimator with lower variance (19).

3 Methods

3.1 Sequential Convolutional Operations for Mean and Variance

We followed the algorithm proposed by the authors in (1) in updating the variational posterior distribution $q_\theta(w|\mathcal{D})$, where we applied two convolutional operations and only one parameter is updated per convolutional operation.

- Treat the output b as an output of a CNN updated by frequentist inference.
  1. Optimize with Adam towards a single point-estimate, and make sure the testing accuracy is increasing
  2. Interpret the single point-estimate as the mean μ_{ijwh} of the variational posterior probability distributions $q(w|\mathcal{D})$
- Learn the variance $\alpha_{ijhw}\mu_{ijhw}^2$ of the variational posterior distribution, where α_{ijhw} is the only parameter needs to be updated

In accordance with the original authors, the activation function we used is Softplus, which is defined as follows:

$$\text{Softplus}(x) = \frac{1}{\beta} \cdot \log(1 + \exp(\beta \cdot x)) \tag{8}$$

By default, $\beta = 1$. Softplus is a smooth approximation to the rectifier, but unlike the wildly and commonly used ReLU function, Softplus function never becomes zero, which is a good property in ensuring that the variance of our variational posterior probability distribution would never become zero.

3.2 Objective Function

The approximated tractable cost function in equation (5), can be derived and explained in greater detail as follows after taking equation (6) into account:

1. Variational Posterior
$$q_\theta(w^{(i)}|\mathcal{D}) = \prod_i \mathcal{N}(w_i|\mu, \sigma^2) \tag{9}$$
$$\Rightarrow \log(q_\theta(w^{(i)}|\mathcal{D})) = \sum_i \log \mathcal{N}(w_i|\mu, \sigma^2) \tag{10}$$
2. Prior
$$p(w^{(i)}) = \prod_i \mathcal{N}(w_i|0, \sigma_p^2) \tag{11}$$
$$\Rightarrow \log(p(w^{(i)})) = \sum_i \log \mathcal{N}(w_i|0, \sigma_p^2) \tag{12}$$
3. Likelihood
$$\log(p(\mathcal{D}|w^{(i)})) \tag{13}$$

3.3 Datasets

We reproduced the model based on the idea of (1), and we were motivated to explore the adaptation and performance of our network on image classification tasks with the MNIST, CIFAR-10, and CIFAR-100 datasets. The first two datasets are often considered small datasets in terms of the number of labeled classes and sample size, while the last one is normally considered as a large dataset.

3.3.1 MNIST

The MNIST database of handwritten digits is a subset of a larger set available from NIST, having 60,000 examples in training set, and 10,000 examples in test set. The digits have been size-normalized and centered in a fixed-size image (20).

3.3.2 CIFAR-10, CIFAR-100

The CIFAR-10 and CIFAR-100 are labeled subsets of the 80 million tiny images dataset, which were collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton (21). The former consists of 60,000 color images of size 32×32 divided into 10 classes, with 50,000 training images and 10,000 test images. They are both widely used datasets and are very much alike in nature, except that the later has 100 classes with 600 images within each, and split into 500 training images and 100 validation images per class.

3.4 Final Model

Given limited time and computational resources, we selected LeNet (22) as our network architecture in testing for both Bayesian and frequentist approach. The architecture of our model is summarized in table 1 below.

layer type	width	stride	padding	input shape	nonlinearity
convolution(5x5)	6	1	0	M x 1 x 32 x 32	Softplus
max-pooling(2x2)		2	0	M x 6 x 28 x 28	
convolution(5x5)	16	1	0	M x 1 x 14 x 14	Softplus
max-pooling(2x2)		2	0	M x 16 x 10 x 10	
fully-connected	120			M x 400	Softplus
fully-connected	84			M x 120	Softplus
fully-connected	10			M x 84	

Table 1: Modified LeNet Architechture for Our Study (1)

4 Results and Discussion

4.1 Model Results

Under Bayesian approach, the training accuracy and test accuracy achieved after 200 epochs by our model for classifying images from MNIST dataset is around 99.10% and 98.23% respectively. The test accuracy was close to that of the original model which achieved 98%(1). We tested the same architecture, i.e. the LeNet without Bayesian inference and get training accuracy of 99.98% and validation accuracy 98.97%. The training and validation processes are presented in the Figure 1 below.

From Figure 1, 2, and 3, we may discover that incorporating Bayesian inference into CNN makes the training process slower, as the accuracies for both training and validation all starts out at lower levels compared to the frequentist approach. We need to make initialization for the mean and variance parameters in the variational posterior probability distribution, and bad initialization values may result in longer time to converge to the optimum solution. The accuracies for the CIFAR-100 dataset manifested the problem even obviously, as the training process showed a plateau pattern at the beginning.

To make a comprehensive comparison, we summarized the accuracies generated in our experiments, and compared to those of the original papers (1) in table 2. As the complexity of the underlying data increases, our model generate both lower training and validation accuracies under Bayesian approach compared to the results of the original paper. However, we can nevertheless observe from our results that the frequentist LeNet approach have large gaps between training and validation error, which is a possible sign of overfitting. While the validation accuracies are lower with Bayesian approach under our experiments, they exhibit robustness against overfitting.

	MNIST	CIFAR-10	CIFAR-100
Our Bayesian LeNet (VI)	98.23 (99.10)	53.51(55.30)	23.13 (26.44)
Our Frequentist LeNet	98.97 (99.98)	65.02(74.95)	33.89 (42.91)
Original Paper Bayesian LeNet (VI)	98	69	31
Original Paper Frequentist LeNet	98	68	33

Table 2: **Validation & Training Accuricies:** Comparison of validation accuricies, along with training accuricies listed in the parenthesis between different architectures and dataset. All values are in percentage.

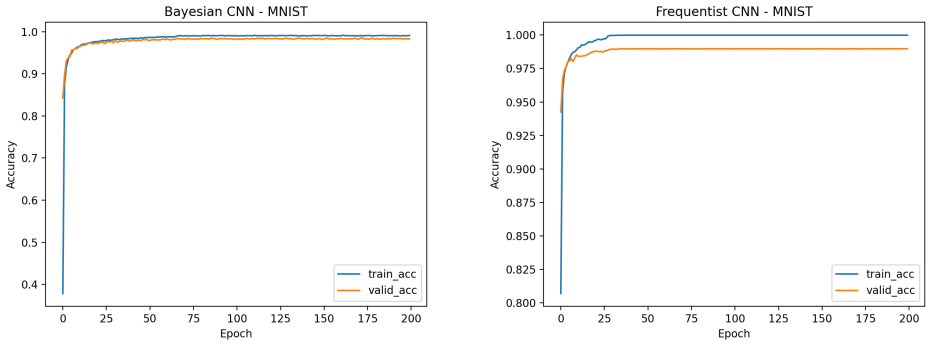

Figure 1: Accuracies of Bayesian LeNet and Frequentist LeNet on MNIST dataset

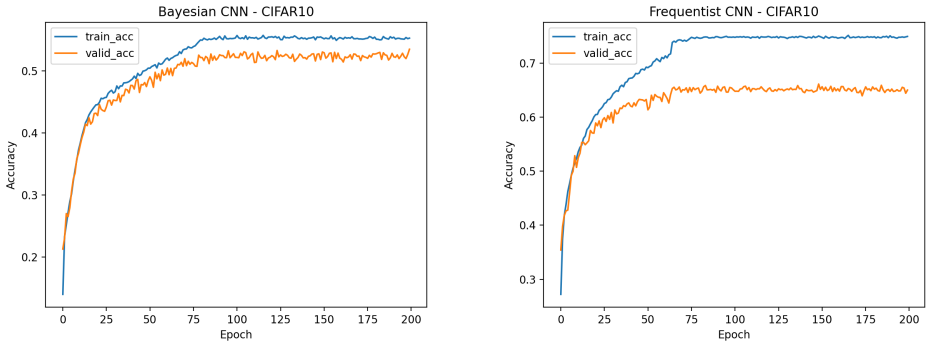

Figure 2: Accuracies of Bayesian LeNet and Frequentist LeNet on CIFAR-10 dataset

4.2 Discussion

This work was limited by computation power and time to test for different architectures. There are several further efforts that can be made to improve the results.

- Since the algorithm includes parameters initialization for the variational posterior probability distribution, we can make further attempt to discover methods in generating good

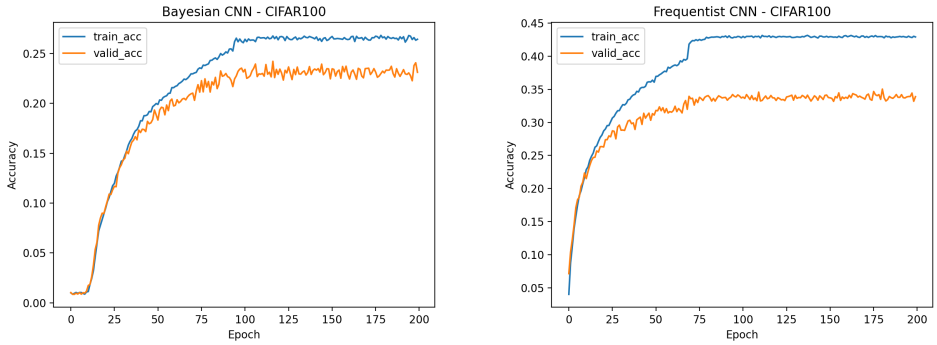

Figure 3: Accuracies of Bayesian LeNet and Frequentist LeNet on CIFAR-100 dataset

initialization values. It is worth discussing to how much extent these randomly initialized values can affect the convergence rate of a Bayesian CNN model.

- The original paper Incorporated variational inference with several CNN architectures, such as AlexNet and different drop out measures. If the the computation resource permits, it is worthwhile trying in integrating Bayesian approach to other more powerful architectures, such as residual network, which surpassed the human level performance in 2015 ImageNet Challange (3).

Another promising future work is to use different distributions, other than Gaussian, in approximating the variational distribution. Gal et. al in 2016 approximated the intractable posterior with Bernoulli variational distribution and proposed the idea of casting dropout as approximate Bernoulli variational inference under neural networks (8). However, they did not place prior distributions $p(w)$ on the CNN's parameters. As most deep learning problems are data dependent, a neural network with an exponential distribution as prior may potentially solve problems in certain fields.

References

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

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[5] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015.

[6] A. Kristiadi, M. Hein, and P. Hennig, "Being bayesian, even just a bit, fixes overconfidence in relu networks," *arXiv preprint arXiv:2002.10118*, 2020.

[7] R. M. Neal, *Bayesian learning for neural networks*, vol. 118. Springer Science & Business Media, 2012.

[8] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Insights and applications," in *Deep Learning Workshop, ICML*, vol. 1, p. 2, 2015.

[9] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992.

[10] G. E. Hinton and D. Van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13, 1993.

[11] A. Graves, "Practical variational inference for neural networks," *Advances in neural information processing systems*, vol. 24, pp. 2348–2356, 2011.

[12] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *arXiv preprint arXiv:1505.05424*, 2015.

[13] M. Fortunato, C. Blundell, and O. Vinyals, "Bayesian recurrent neural networks," *arXiv preprint arXiv:1704.02798*, 2017.

[14] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*, pp. 355–368, Springer, 1998.

[15] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on information theory*, vol. 51, no. 7, pp. 2282–2312, 2005.

[16] K. Friston, J. Mattout, N. Trujillo-Barreto, J. Ashburner, and W. Penny, "Variational free energy and the laplace approximation," *Neuroimage*, vol. 34, no. 1, pp. 220–234, 2007.

[17] L. K. Saul, T. Jaakkola, and M. I. Jordan, "Mean field theory for sigmoid belief networks," *Journal of artificial intelligence research*, vol. 4, pp. 61–76, 1996.

[18] T. S. Jaakkola and M. I. Jordan, "Bayesian parameter estimation via variational methods," *Statistics and Computing*, vol. 10, no. 1, pp. 25–37, 2000.

[19] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," *Advances in neural information processing systems*, vol. 28, pp. 2575–2583, 2015.

[20] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," 2010.

[21] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.