[Re] VARIATIONAL BAYESIAN LAST LAYERS

Sanghoon Kim

Department of Data Science, 24510101

shkim@ds.seoultech.ac.kr

Taegyeom Bae

Department of Mechanical Information Engineering, 23512080

skygarden058@seoultech.ac.kr

Jeongmin An

Department of Mechanical Information Engineering, 24510091

asekooan1@seoultech.ac.kr

Reproducibility Summary

Scope of Reproducibility

The original paper introduces Variational Bayesian Last Layers (VBLLs), a novel last layer neural network component. The authors claim that this approach yields a sampling-free, single-pass model and loss that effectively improves uncertainty estimation.

This paper verifies these claims and analyzes the algorithm across various datasets, hyperparameters, and combinations with other algorithms providing further understanding and insights.

Methodology

Based on the claims of the paper, we focused on reproducing two main results: classification and regression.

For classification, we used the MNIST dataset, which is different from the original paper, to replicate the results. We also varied the learning rate, batch size, weight decay, and epochs to observe the impact on the experiment's outcomes.

For regression, we created a simple dataset to investigate the role of the KL Penalty, a key parameter of VBLL. Additionally, we extended the method proposed in the paper by combining it with other techniques to explore performance improvements.

All subsequent work was conducted based on adaptations of the code provided by the authors.

Results

In the classification task, we confirmed that applying different datasets resulted in outcomes that differed from those claimed by the original paper's authors. We also presented and analyzed how various hyperparameters influence the results. Particularly, we found that weight decay and learning rate significantly affect the VBLL algorithm, providing new insights to our readers.

In the regression task, we provided an intuitive and visual understanding of VBLL and examined the impact of its key parameter, the KL Penalty. This generally aligned with the claims of the original paper's authors. Furthermore, we proceeded with combining VBLL with other algorithms as suggested in the original paper, visually presented and analyzed these results, indicating possibilities for subsequent research and various applications.

What was easy

The authors provided the algorithm in a packaged format, eliminating the need for intricate re-implementation. The core idea of applying variational methods to the last layer of a neural network is straightforward and novel, leading to relatively simple reproduction code.

33rd Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

What was difficult

Understanding the content of the paper was initially challenging due to its advanced concepts. Despite having basic knowledge of KL divergence and Bayesian theory, the concept of Bayesian Neural Networks was new to us.

Also, deciding which parts to reproduce and how to make it beneficial for readers required careful consideration. At first, we aimed to extend the code provided by the authors to compare the performance with various existing techniques.

However, instead, we focused on utilizing the given code to provide insights into the algorithm. We analyzed the impact of one of the core components, the KL Penalty, and observed the results by varying datasets and hyperparameters. Additionally, we implemented and analyzed the integration of VBLL with other networks.

The original paper mentions the necessity of these analyses but does not provide actual results. Our findings offer additional research opportunities and insights for readers regarding the paper's contributions.

Communication with original authors

There was no communication with the original authors; the project was carried out by adapting the code provided by the authors.

1 Introduction

Bayesian Last Layer (BLL) models are designed to enhance neural networks by incorporating Bayesian inference in the final layer to improve uncertainty quantification. BLL models typically aim to optimize the log marginal likelihood via gradient descent.

$$T^{-1} \log p(Y \mid X, \theta) \tag{1}$$

Many existing BLL algorithms can accurately compute the marginal likelihood, allowing them to handle uncertainty efficiently and offering lower variance. However, these methods often suffer from high computational costs and biased gradient estimates.

To address these issues, the original paper introduces VBLL, a new last layer for uncertainty quantification. Instead of directly calculating the full marginal likelihood, VBLL employs Stochastic Variational Inference (SVI) to approximate the marginal likelihood. The objective is to maximize the evidence lower bound (ELBO) on the marginal likelihood:

$$T^{-1} \log p(Y \mid X, \theta) \geq L(\theta, \eta, \Sigma) - T^{-1}\mathrm{KL}(q(\xi \mid \eta) \| p(\xi)). \tag{2}$$

By maximizing the right-hand side of equation (2), VBLL performs efficient Bayesian inference and optimizes both the network weights and uncertain parameters. This approach ensures effective uncertainty quantification with reduced computational overhead.

2 Scope of reproducibility

The original paper investigates the application of Variational Bayesian Last Layers (VBLL) by conducting experiments divided into classification and regression tasks.

For classification tasks, the original paper uses the SVHN and CIFAR-100 datasets and compares the results with other baseline models. It concludes that the Discriminative VBLL (D-VBLL) method shows superior performance in accuracy, error rate, and out-of-distribution (OOD) detection, followed by the Generative VBLL (G-VBLL) and other models. We aim to validate these claims by replicating the experiments on a different dataset, the MNIST dataset. Our replication results show differing outcomes from those presented in the original paper.

Additionally, the paper suggests that VBLL model performance is influenced by typical neural network hyperparameters such as learning rate and batch size, among other parameters. Although the paper mentions the necessity for a hyperparameter search, it does not discuss the outcomes of such comparisons. Therefore, we will also explore how varying these hyperparameters affects the performance of the three models, D-VBLL, G-VBLL, and MLP models.

Regarding regression tasks, the original paper evaluates the performance of VBLL using UCI datasets and compares it against other baseline models. Instead of merely reproducing these results, our study investigates the effect of variations in the KL Penalty, a critical parameter of VBLL. Also, the original paper indicates that an increase in the KL Penalty strengthens regularization and enhances generalization performance. We plan to reproduce these findings using a simple dataset and provide visual illustrations through graphs for intuitive understanding.

Furtermore, we extend the exploration by implementing a combination not covered in the original paper, specifically the integration of VBLL with the Spectral Normalized Neural Gaussian Process (SNGP). We will analyze the impact of this integration on regression tasks, assessing how the combination influences performance.

All these replication results are detailed in Section 4.

3 Methodology

In the classification task, we replicate the results of the original paper using the MNIST dataset. Specifically, we utilize a simple MLP and two types of VBLL introduced in the original paper: D-VBLL and G-VBLL. These three algorithms are evaluated on three key metrics: loss, error rate, and out-of-distribution (OOD) detection. Additionally, we examine the effects of various hyperparameter modifications to provide new insights into VBLL to our readers. The conditions for the hyperparameter experiments are as follows:

Base Condition:

- Learning Rate = 1×10^{-3}
- Batch Size = 512
- Weight Decay = 1×10^{-4}
- Number of Epochs = 30

High Learning Rate:

- Learning Rate = 1×10^{-2} **(increased)**
- Batch Size = 512
- Weight Decay = 1×10^{-4}
- Number of Epochs = 30

Low Batch Size:

- Learning Rate = 1×10^{-3}
- Batch Size = 256 **(decreased)**
- Weight Decay = 1×10^{-4}
- Number of Epochs = 30

High Batch Size:

- Learning Rate = 1×10^{-3}
- Batch Size = 1024 **(increased)**
- Weight Decay = 1×10^{-4}
- Number of Epochs = 30

High Weight Decay:

- Learning Rate = 1×10^{-3}
- Batch Size = 512
- Weight Decay = 1×10^{-3} **(increased)**
- Number of Epochs = 30

Long Training:

- Learning Rate = 1×10^{-3}
- Batch Size = 512
- Weight Decay = 1×10^{-4}
- Number of Epochs = 50 **(extended)**

In the regression tasks, we first generate a simple dataset to visually demonstrate the performance of a simple MLP model and VBLL, facilitating intuitive understanding. We then visualize the results of three variations of VBLL's key parameter, the KL Penalty (10X, 25X, 50X), to provide a visual and intuitive grasp of its impact on VBLL.

The VBLL algorithm can be simply applied to the final layer, which makes it highly compatible with other deep learning algorithms. The original paper suggests this as a direction for further research. So, we combine SNGP with VBLL intuitively demonstrate its performance in regression compared to simple VBLL, thus providing new insights for subsequent research.

All reimplementation tasks are conducted on Google Colab under TPU environments. For the classification task, training the algorithms under basic conditions takes about 20 minutes, while the regression tasks can be completed in a few minutes. All reimplementation codes are available on the author's GitHub (https://github.com/SHKim26/VARIATIONAL_BAYESIAN_LAST_LAYERS).

4 Results

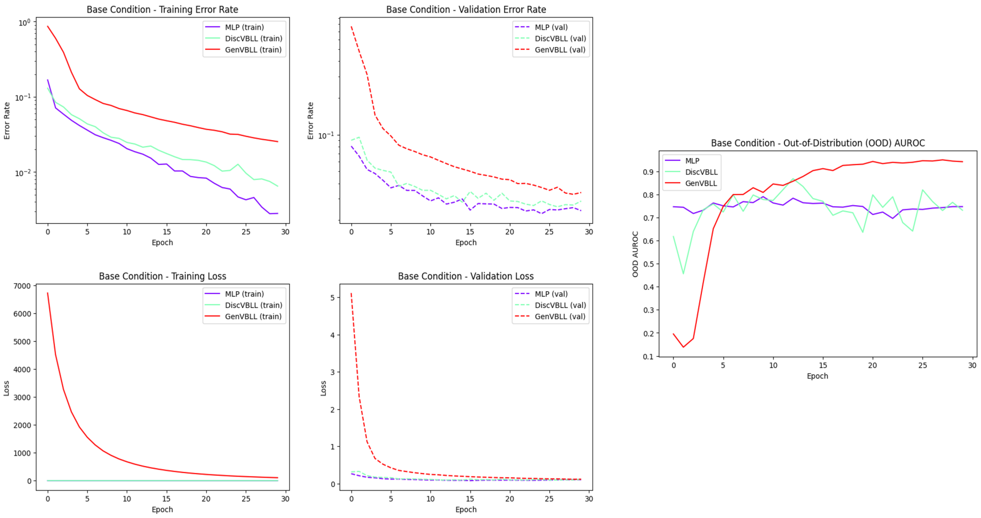
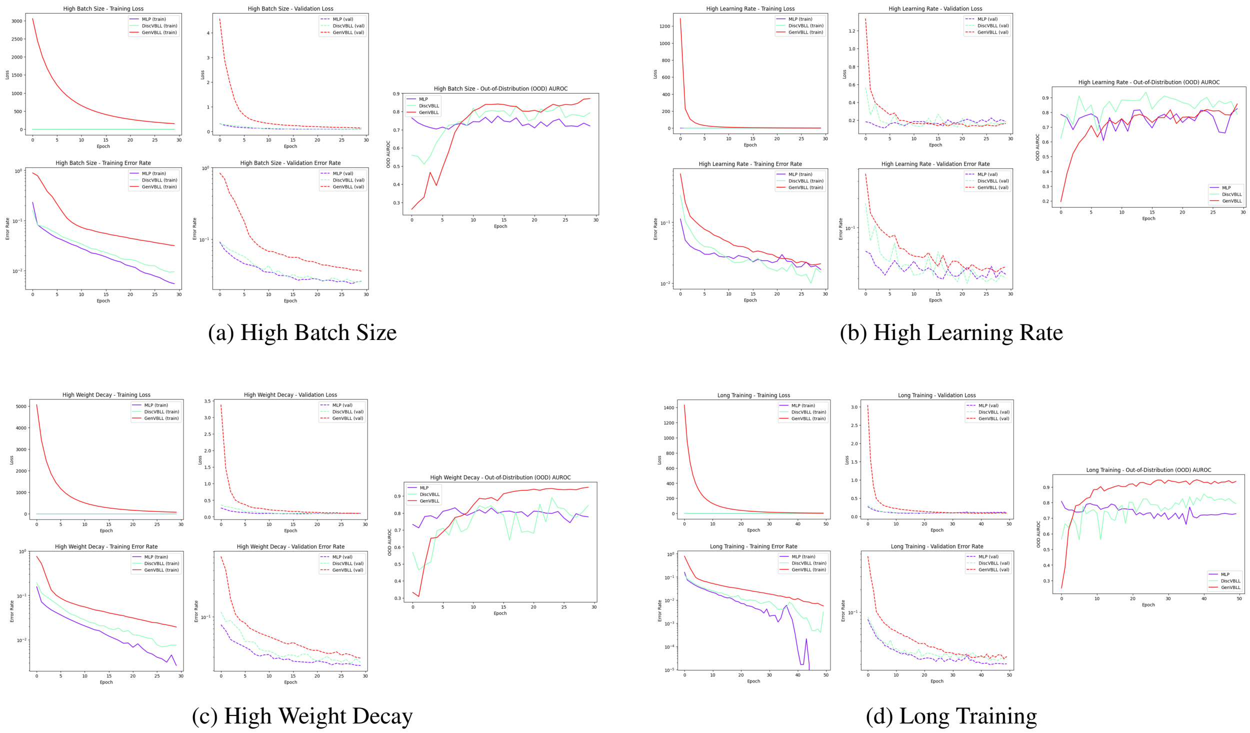


Figure 1: Result of Base Condition.

Figure 1 displays graphs showing the results of applying MLP, D-VBLL, and G-VBLL to the MNIST dataset. The original paper claims that in the classification task, G-VBLL shows the best results, followed by D-VBLL. Although there are minor differences in hyperparameters from the original paper, the outcomes varied when reimplemented with a different dataset. Notably, G-VBLL performed best in out-of-distribution (OOD) scenarios, while D-VBLL showed almost no difference from the simple MLP.



(a) High Batch Size

(b) High Learning Rate

(c) High Weight Decay

(d) Long Training

Figure 2: Results of Various Hyperparameters

Figure 2 presents graphs of the results for various hyperparameters introduced in Section 3. By experimenting with different hyperparameters of VBLL, we provide new insights to our readers, with notable results as follows:

- a) Increasing the batch size, like other deep learning algorithms, slowed down the learning process, which is easily observable in the loss and OOD graphs. A batch size of 1024 is not suitable for datasets like MNIST, as clearly demonstrated by the results.
- b) Increasing the learning rate showed that D-VBLL, consistent with the original paper, performs best in OOD scenarios. Interestingly, all three algorithms exhibited significant oscillations in the error rate and loss graphs when the learning rate was increased. While the overall trend remains unchanged, this suggests that the learning process is somewhat unstable.
- c) Increasing weight decay resulted in a significant destabilization and deterioration in the OOD performance of D-VBLL, showing worse results than the simple MLP.
- d) Extending the training duration by increasing the number of epochs revealed that D-VBLL's training error rate spikes around epoch 45, suggesting potential overfitting. In terms of OOD results, G-VBLL, which was trained for an extended period, exhibited the best performance on the MNIST dataset, clearly differing from the outcomes reported in the original paper.

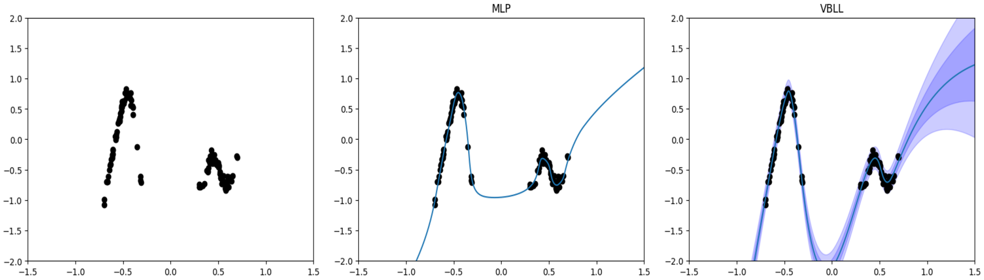


Figure 3: Comparison of MLP and VBLL

In regression tasks, we initially generate a simple dataset as shown on the left side of Figure 3 to visually and intuitively understand VBLL. For a basic MLP, as depicted in the middle of Figure 3, the learning outcomes follow the data points closely with a blue prediction line; however, this does not quantitatively express the uncertainty or variability of the predictions. The VBLL model depicted in the right area of Figure 3 not only predicts as accurately as the MLP, but it also quantifies the uncertainty of predictions at each point along the prediction line. This shaded confidence area is very small at data points and increases in regions with fewer data points or greater variability.

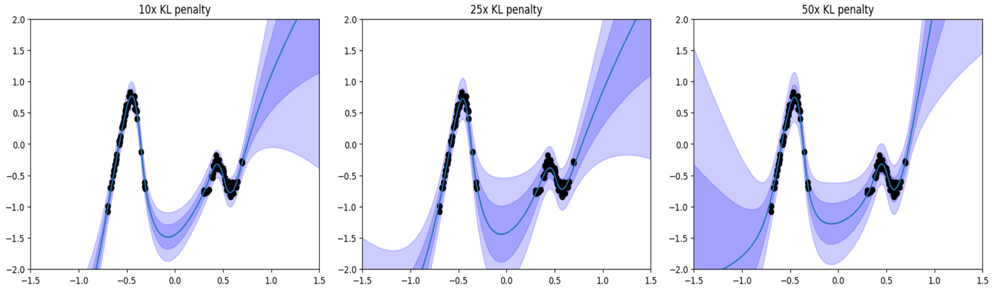


Figure 4: Results of Varying KL Penalty

Figure 4 illustrates the impact of one of the key parameters of the VBLL algorithm, the KL Penalty. As the KL Penalty increases, the area representing the uncertainty of the predictions grows larger. Additionally, it is observable that the blue line representing the predictions is gradually distancing from the data points. This aligns with the paper's assertion and experimental results that increasing the KL Penalty tends to prevent overfitting and assess uncertainty more significantly.

When applying VBLL, it is crucial to set an appropriate KL Penalty depending on the application area. Care must be taken not to set the KL Penalty too high, which could lead to ignoring essential features of the data and overly generalizing.

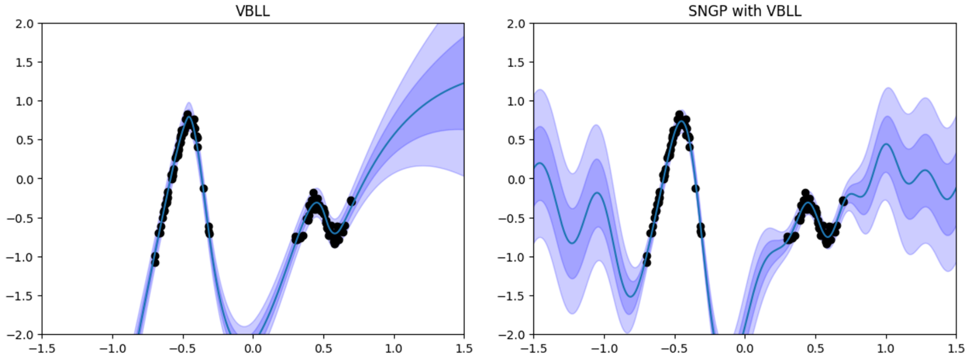


Figure 5: Results of SNGP with VBLL

Figure 5 compares the results of the VBLL model and its combination with the SNGP model. SNGP enhances network predictability by applying residual connections, spectral normalization, and Random Fourier Features, effectively managing prediction uncertainty. By integrating VBLL with this model, we examined the performance and predictions shown in combinations with other networks as anticipated by the original paper's authors.

The combination of SNGP and VBLL demonstrates increased levels of uncertainty but effectively captures and reflects nonlinear patterns of the data in its predictions. The SNGP with VBLL model aligns closely with actual data points, showing minimal uncertainty in areas with substantial data and effectively quantifying uncertainty in regions with sparse data points by capturing the nonlinear patterns of the data.

5 Conclusion

This paper re-implements and verifies the results of the VBLL, providing new insights through additional experimental results.

We applied the VBLL algorithm in a classification task using different datasets and varying hyperparameters, revealing that results can differ from the claims of the original paper depending on the dataset used. Notably, we identified the critical need to carefully adjust the weight decay and learning rate values when applying VBLL to the classification task. This finding is crucial when employing VBLL across various datasets or when conducting further research.

In regression tasks, we provided an intuitive understanding of VBLL's key parameter, the KL Penalty, and assessed its impact on VBLL. Furthermore, by combining VBLL with other algorithms such as SNGP, we visually demonstrated and analyzed how such integrations affect VBLL's performance, presenting additional research and application possibilities. Our results enhance the understanding of VBLL, proposing various applications and further research opportunities, thereby aiding future researchers in better utilizing this method. The intuitive understanding of the original paper and all reimplementation code are provided on the author's GitHub.

References

[1] J. Harrison, J. Willes, and J. Snoek, "Variational bayesian last layers," *arXiv preprint arXiv:2404.11599*, 2024.

[2] K. Azizzadenesheli, E. Brunskill, and A. Anandkumar, "Efficient exploration through bayesian deep q-networks," in *2018 Information Theory and Applications Workshop (ITA)*. IEEE, 2018, pp. 1–9.

[3] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, 2013.

[4] J. Knoblauch, J. Jewson, and T. Damoulas, "Generalized variational inference: Three arguments for deriving new posteriors," *arXiv preprint arXiv:1904.02063*, 2019.

[5] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International conference on machine learning*. PMLR, 2015, pp. 1613–1622.

[6] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams, "Scalable bayesian optimization using deep neural networks," in *International conference on machine learning*. PMLR, 2015, pp. 2171–2180.

[7] J. Harrison, A. Sharma, and M. Pavone, "Meta-learning priors for efficient online bayesian regression," in *Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics 13*. Springer, 2020, pp. 318–337.

[8] M. Collier, B. Mustafa, E. Kokiopoulou, R. Jenatton, and J. Berent, "Correlated input-dependent label noise in large-scale image classification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1551–1560.