

FastBNNs: Fast training and inference of Bayesian neural networks

David J. Schodt ¹

¹ Los Alamos National Laboratory, USA

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: 

Submitted: 21 October 2025

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

Neural networks (NNs) are a flexible class of models that can be used to approximate complicated functions. Bayesian neural networks (BNNs) extend NNs by treating their learnable parameters as distributions, enabling uncertainty quantification of both model outputs and of the parameters themselves. FastBNNs defines a PyTorch-based (Paszke et al., 2019) framework for BNN training and inference and implements a set of recently developed algorithms for fast and flexible BNN inference (Schodt et al., 2024; Schodt, 2024). Using thin wrappers around PyTorch modules and tensors, FastBNNs enables one-line conversion of existing neural network architectures to their Bayesian counterparts. FastBNNs was designed to simplify and accelerate the adoption of BNNs in NN applications that benefit from uncertainty quantification.

Statement of need

In many applications, NNs are overparameterized black-box models that can confidently produce erroneous predictions (Blundell et al., 2015). Principled approaches to uncertainty quantification are thus highly desirable for NNs, as they provide not only predictive uncertainties in model outputs but also parameter uncertainties that can drive advanced training strategies like model pruning. BNNs treat NN parameters as distributions and hence naturally provide the desired uncertainty quantification. Despite decades of research in BNNs, available software implementations require redefining NNs in a bespoke framework (Bingham et al., 2019), are restricted to a limited set of layers (Esposito, 2020; Krishnan et al., 2022), and/or do not model heteroscedastic uncertainty in the data (Daxberger et al., 2021).

FastBNNs provides a Python implementation of a set of recently developed algorithms for approximate Bayesian inference (Schodt et al., 2024; Schodt, 2024) that support variational inference of BNNs assuming a mean-field approximation (i.e., independent parameters). By wrapping existing PyTorch-based models, FastBNNs enables fast and flexible BNN training and inference without sacrificing the simplicity, flexibility, and familiarity of PyTorch. FastBNNs fills a gap in existing open-source BNN software by simplifying NN-to-BNN conversion and leveraging fast inference algorithms, encouraging further research in and adoption of BNNs.

References

- Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P. A., Horsfall, P., & Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20, 28:1–28:6. <http://jmlr.org/papers/v20/18-403.html>
- Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural network. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd international*

- 39 conference on machine learning (Vol. 37, pp. 1613–1622). PMLR. <https://proceedings.mlr.press/v37/blundell15.html>
- 40
- 41 Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., & Hennig, P. (2021).
42 Laplace redux—effortless Bayesian deep learning. *NeurIPS*.
- 43 Esposito, P. (2020). BLiTZ - bayesian layers in torch zoo (a bayesian deep learning library for
44 torch). In *GitHub repository*. <https://github.com/piEsposito/blitz-bayesian-deep-learning/>;
45 GitHub.
- 46 Krishnan, R., Esposito, P., & Subedar, M. (2022). *Bayesian-torch: Bayesian neural network*
47 *layers for uncertainty estimation*. <https://doi.org/10.5281/zenodo.5908307>
- 48 Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z.,
49 Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M.,
50 Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An
51 imperative style, high-performance deep learning library. In *Advances in neural information*
52 *processing systems 32* (pp. 8024–8035). Curran Associates, Inc. [http://papers.neurips.cc/](http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf)
53 [paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf](http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf)
- 54 Schodt, D. J. (2024). *Few-sample Variational Inference of Bayesian Neural Networks with*
55 *Arbitrary Nonlinearities*. <https://arxiv.org/abs/2405.02063>
- 56 Schodt, D. J., Brown, R., Merritt, M., Park, S., Menolascino, D., & Peot, M. A. (2024).
57 *A Framework for Variational Inference of Lightweight Bayesian Neural Networks with*
58 *Heteroscedastic Uncertainties*. <https://arxiv.org/abs/2402.14532>

DRAFT