A Simulation-based Framework for Characterizing Predictive Distributions for Deep Learning

Jessica Ai [1] Beliz Gokkaya [1] Ilknur Kaynar Kabul [1] Audrey Flower [1] Ehsan Emamjomeh-Zadeh [1] Hannah Li [1] Li Chen [1] Neamah Hussein [1] Ousmane Dia [1] Sevi Baltaoglu [1] Erik Meijer [1]

Abstract

Characterizing the confidence of machine learning predictions unlocks models that know when they do not know. In this study, we propose a framework for assessing the quality of predictive distributions obtained using deep learning models. The framework enables representation of aleatory and epistemic uncertainty, and relies on simulated data to generate different sources of uncertainty. Finally, it enables quantitative evaluation of the performance of uncertainty estimation techniques. We demonstrate the proposed framework with a case study highlighting the insights one can gain from using this framework.

1. Introduction

While we rely on deep neural networks (DNNs) to make decisions in a wide variety of applications, they do not provide information on the confidence of their predictions. However we can make better-informed decisions if we understand the trustworthiness of model predictions. For instance, predictive uncertainty can help identify when DNNs fail in the presence of adversarial (Smith & Gal, 2018) or out-of-distribution inputs (Snoek et al., 2019). There are also many applications of using such information, such as in bandit settings for recommendation systems (Thompson, 1933) and reinforcement learning (Dabney et al., 2018).

To this end, Bayesian neural networks (BNNs) (Neal, 2012) allow us to estimate predictive uncertainty. Unfortunately at the scale in which deep learning is applied in industry, Bayesian inference is generally intractable. In recent years, alternative approximations of the Bayesian posterior have been proposed such as the Monte Carlo simulation based application of dropout (Gal & Ghahramani, 2016), variance networks (Neklyudov et al., 2018) or partial Bayesian approaches such as Bayesian last layer (Snoek et al., 2015). Past assessments of the merits of these methods typically rely on gains in predictive accuracy, via the mean of the predictive distribution. However, we can also make use of other characteristics of the distribution when evaluating risk-return tradeoffs for decision making, and it is therefore crucial to assess its full distributional properties.

In this study, we propose a simulation based framework that enables the comparison of distributional information across different uncertainty estimation techniques. It is inspired by BNNs and is generalizable to any model architecture. By controlling the amount of uncertainty added to a model and dataset, our framework permits comparisons with respect to the uncertainty injected into the system. We first outline the framework, focusing on how different sources of uncertainty can be simulated in this setting. This is followed by a case study where we demonstrate, with a toy dataset, the possible insights one can obtain through this framework.

2. The Framework

2.1. Controlling Different Sources of Uncertainty

For a model $\boldsymbol{f}(\cdot)$ characterized by parameters $\boldsymbol{\theta}$, the target variable $\boldsymbol{y}$ is represented using input features $\boldsymbol{x}$ as:

$$\boldsymbol{y} = f(\boldsymbol{x};\boldsymbol{\theta}) + \boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{\epsilon}$ represents the residuals in predictions.

Sources of uncertainty are commonly classified into epistemic and aleatory uncertainty, where the former is defined as uncertainty due to limited knowledge about the data generating process and the latter stems from inherent randomness in the data (Tagasovska & Lopez-Paz, 2019). Following a BNN representation, epistemic uncertainty can be represented in model parameters as:

$$\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} + \boldsymbol{\epsilon_\theta} \tag{2}$$

where $\boldsymbol{\epsilon_\theta}$ represents the noise we add to the fitted model, characterized here by the parameters θ. In our framework, $\boldsymbol{\epsilon_\theta}$ can be configured by users and we support adding noise to individual parameters of the model independently as well as assuming a covariance structure.

[1] Facebook. Correspondence to: Jessica Ai <jaix@fb.com>, Beliz Gokkaya <belizg@fb.com>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning. Copyright 2020 by the author(s).

We also incorporate aleatory uncertainty through mimicking measurement error through similarly adding noise $\boldsymbol{\epsilon_x}$ to the input features as well as $\boldsymbol{\epsilon_y}$ to the target variables. Then by treating sources of uncertainty independently, we can generate input features and target variables to be used with uncertainty estimation techniques via the following process:

$$\begin{aligned}\tilde{\boldsymbol{x}} &= \boldsymbol{x} + \boldsymbol{\epsilon_x} \\ \tilde{\boldsymbol{y}} &= f(\boldsymbol{x}; \tilde{\boldsymbol{\theta}}) + \boldsymbol{\epsilon_y}\end{aligned} \tag{3}$$

Since we control the level of uncertainty injected into the system, we can speculate on the uncertainty we should retrieve when we estimate the uncertainty of the dataset $\{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}\}$. This enables us to perform quantitative analysis by comparing an estimated distribution with the empirical predictive distribution, formed from Monte Carlo samples of our simulation process. Evidently, the simulations heavily depend on the original model, so we note that they will be limited by the model's structure.

2.2. Evaluating with Simulated Uncertainty

Previously, predictive accuracy metrics such as Mean Squared Error (MSE) based on the average of the predictive samples (Gal & Ghahramani, 2016; Hernández-Lobato & Adams, 2015) and coverage probabilities (Romano et al., 2019) have been used for quantitative evaluations with respect to dataset labels. Snoek et al. (2019) also performed an indirect evaluation by measuring the accuracy of a classifier trained on the task of detecting out-of-distribution inputs with uncertainty estimates. There have also been qualitative evaluations that check whether epistemic uncertainty increases after removal of input regions (Kendall & Gal, 2017) or that there exists a positive correlation between uncertainty and prediction error (McMahan et al., 2013).

While these are all valuable and necessary evaluations, we believe that additional evaluations with respect to the ground truth predictive distribution from our simulation procedure will paint a more complete picture. Here we briefly discuss examples of such comparisons. First, we can directly evaluate the full predictive distribution using measures such as Wasserstein distance. We can also evaluate specific characteristics of the distribution. For instance, there may be applications where we are only interested in confidence intervals or techniques where we directly estimate confidence intervals (Romano et al., 2019). For this purpose, we introduce our own confidence interval accuracy metrics based on classification accuracy metrics, which we will refer to as $CI_{P(recision)}$ and $CI_{R(ecall)}$, as illustrated in Figure 1 and defined in Equations 4 and 5.

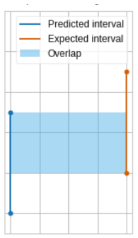


Figure 1. Illustration of the confidence interval based accuracy metrics

$$CI_P = \frac{CI_{Width_{overlap}}}{CI_{Width_{expected}}} \tag{4}$$

$$CI_R = \frac{CI_{Width_{overlap}}}{CI_{Width_{predicted}}} \tag{5}$$

3. Case Study

In this case study, we illustrate how the aforementioned simulation and evaluation procedure can be used to obtain insights about predictive distributions estimated from different techniques. We note that this study by no means provides an exhaustive analysis of these techniques.

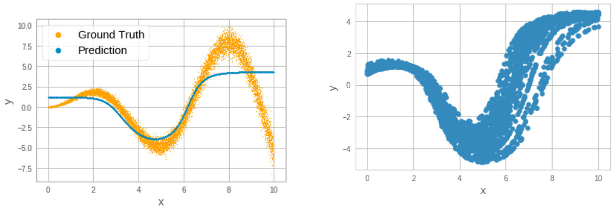
3.1. Toy regression model and dataset

For this case study, we use a toy sinusoidal dataset of the form:

$$y = x\ \sin(x) + 0.1\ x\ \epsilon \tag{6}$$

where $x \sim$ Uniform(0, 10) and $\epsilon \sim$ Normal(0, 1).

This dataset is then used to train a four layer feedforward network, where the predictions are shown in Figure 2a.



(a) Toy regression dataset

(b) Toy regression dataset with added model parameter noise

Figure 2. Case study dataset

3.2. Uncertainty Estimation Methods

We briefly overview the techniques we will employ, which are largely those that estimate epistemic uncertainty. These methods are implemented in PyTorch (Paszke et al., 2019), and make heavy use of broadcasting for vectorization of computations that require repeated sampling. For all, we use an MSE based loss criterion with 100 training epochs..

3.2.1. Monte Carlo Dropout

Dropout is frequently used for regularization and Gal & Ghahramani (2016) proposed using dropout at inference time following a Monte-Carlo (MC) approach to estimate predictive distributions. It has been shown that this provides an approximation to the Bayesian posterior. For the case study, we add dropout layers after each fully connected layer with a dropout probability of 0.1. We use this model with 100 samples at inference time to collect empirical samples, yielding the predictive uncertainty.

3.2.2. Bayes by Backprop

Blundell et al. (2015) proposed a variational inference approach to estimate the posterior distribution of the weights of a Bayesian neural network. Here, we use this approach with standard Gaussian priors on the model parameters with 100 samples drawn from the posterior distribution to represent the predictive distribution.

3.2.3. Bootstrapping

Bootstrapping is a randomization based ensembling technique for approximating the predictive distribution (Paass, 1993). In this study, we adopt a nonparametric bootstrapping approach where we repeatedly perform sampling with replacement. For each bootstrap sample set, we train a model and study the predictive distribution across all datasets. We use this model with 100 bootstrap samples.

3.2.4. Ensembling

A randomization-based model ensembling approach is proposed for deep neural networks in Lakshminarayanan et al. (2017) to estimate predictive uncertainty. Similar to this approach, we train an ensemble of models where we use ten random initialization of model parameters.

3.3. Estimating Uncertainty of the Original Dataset

Before we apply the framework, we first review the uncertainty estimation techniques on the original dataset. We investigate predictive accuracy improvements that can be gained using the distributional information in an average sense in Table 1.

Method	Coverage	$\overline{CI_W}$	MSE
MC Dropout	0.88	5.07	1.8
BBB	1.00	8.6	1.1
Bootstrap	0.73	2.18	3.43
Ensemble	0.49	1.52	0.61

Table 1. Results using the original dataset. Here we show 95% coverage probabilities and mean 95% confidence interval widths ($\overline{CI_W}$) using the empirical predictive distributions. We also compute MSE of the mean predictive sample.

We observe that while Bayes by Backprop (BBB) provides high coverage, it also has the widest confidence intervals. This signals that it may be overestimating the predictive uncertainty. On the other hand, ensembling has the opposite trend to BBB in that predictive uncertainty is underestimated. Bootstrap has lower coverage compared to BBB but indicates poor predictive accuracy through its high MSE. MC Dropout seems to be providing the best compromise across these metrics.

3.4. Simulation experiments

We will now use the simulation framework to further our understanding of the results in Table 1 by testing the models and predictive distributions under different sources of uncertainty.

3.4.1. Model Parameter Uncertainty

First, we are interested in a simple setting where the true model parameter uncertainty is independently and identically distributed (i.i.d.) following a Gaussian distribution with hierarchical priors. Figure 2b shows the simulated data and Table 2 lists the results obtained in the presence of parameter uncertainty.

On the left portion of Table 2, we observe similar trends amongst the methods to those in Table 1. This time, with the simulation framework, we have access to the true model parameter distributions and can obtain further insights. Looking at the right portion of Table 2, we observe that BBB's intervals have perfect recall but they also possess the lowest precision. MC Dropout shows a similar trend for the average interval precision and recall metrics. However, we observe that there is considerable variability in these metrics for MC Dropout across the dataset. Furthermore, Wasserstein distance for MC Dropout is the second highest, indicating the predictive intervals might be problematic. Indeed, when we plot the expected versus predicted confidence intervals in Figure 3, we observe that the predictive uncertainty deviates from the expected distribution. We also note that this toy experiment is designed to have increased uncertainty as the magnitude of the input feature increases. Only bootstrapping and ensembling are observed to capture this trend

Method	Coverage	$\overline{CI_W}$	MSE	$\overline{CI_P}$	$\overline{CI_R}$	$\overline{d_2}$
MC Dropout	0.83	2.97	0.67	0.51 ± 0.3	0.8 ± 0.23	0.45 ± 0.26
BBB	1.0	8.02	0.73	0.23 ± 0.18	1.0 ± 0	1.26 ± 0.12
Bootstrap	0.53	0.68	0.52	0.99 ± 0.06	0.45 ± 0.17	0.34 ± 0.32
Ensemble	0.41	0.57	0.74	1.0 ± 0	0.27 ± 0.09	0.35 ± 0.32

Table 2. Results with model uncertainty. On the right, we now include metrics enabled by our framework, interval precision (CI_P), interval recall (CI_R) and Wasserstein distance (d_2). We compute these metrics for individual predictions in the simulated dataset and report the mean and standard deviation across the dataset. We can now also compare $\overline{CI_W}$ with the true mean width of 1.85.

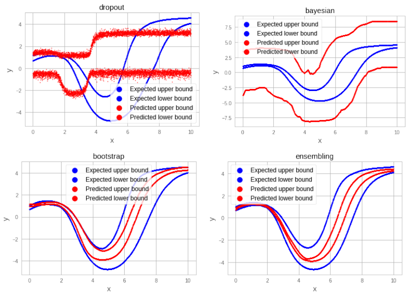


Figure 3. Expected and predicted 95% confidence interval trends under model uncertainty

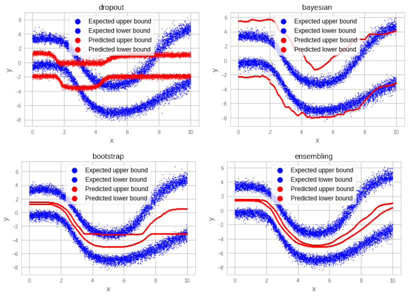


Figure 4. Expected and predicted 95% confidence interval trends under model and data uncertainty

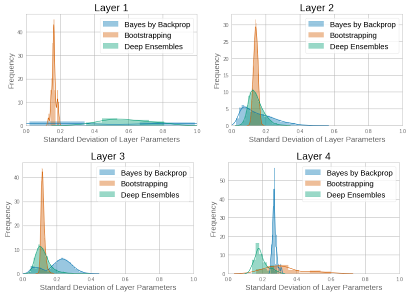


Figure 5. The standard deviation in predicted model parameters varies across the layers of the feedforward network.

despite poor confidence interval recall and accordingly in Table 2, we obtain the lowest Wasserstein distances for these two methods.

3.4.2. Data Uncertainty

In order to investigate how the techniques in our study behave in the presence of aleatory uncertainty, we add standard Gaussian label noise on top of the model parameter noise from the previous section. The results in Table 3 show that there is degradation in metrics for all methods but BBB, which is more resilient to this uncertainty. As before, in Figure 4 we can also visually inspect the 95% confidence bounds for each method where we observe similar trends.

Method	Coverage	$\overline{CI_W}$	$\overline{CI_P}$	$\overline{CI_R}$	$\overline{d_2}$
MC Dropout	0.63	3.1	0.82	0.54	0.77
BBB	0.98	7.75	0.63	0.98	0.71
Bootstrap	0.18	0.7	1.0	0.14	0.93
Ensemble	0.15	0.61	1.0	0.11	1.0

Table 3. Results under model and data uncertainty.

3.4.3. Exploring layerwise applications of uncertainty estimation techniques

One constraint encountered in practice is that applying such techniques to the full model may be too computationally expensive and similar to Snoek et al. (2015), we may only wish to apply these methods to particular layers. We can also use our simulation framework to better understand this tradeoff. First, we can further inspect the trained model weights for each layer of the feedforward network to understand how uncertainty was learnt in its parameters. We show results in Figure 5 for the previous simulation setting, where it is clear that even though we introduced uncertainty in an i.i.d. fashion to all model parameters in the simulation setting, the uncertainty learnt by each model can vary immensely from layer to layer. We do not show results for MC Dropout because it enables learning different model substructures rather than different model parameter values.

We show the results obtained by applying partial Bayesian layers in Table 4. As expected, we observe that the training time is improved significantly by using a partial approach. The results deviate from a full Bayesian approach as we remove Bayesian layers with the closest results obtained using the last few layers, which according to Figure 5 have the highest variability. Together with the results on the original dataset, these additional insights gained through the simulation framework can help users make better informed decisions when selecting a particular technique.

Bayesian Layer	Coverage	$\overline{CI_W}$	$\overline{CI_P}$	$\overline{CI_R}$	$time_{train}$
All	1.0	8.02	0.23	1.0	1.82
First	0.88	3.0	0.65	0.88	0.87
Last Two	0.99	7.0	0.3	0.98	1.17
Last	0.97	5.56	0.37	0.95	0.7

Table 4. Results applying partial Bayesian layers under model and data uncertainty.

4. Conclusion

With the increasing interest in characterizing predictive distributions of DNNs, we propose a simulation-based framework that is simple and generalizable to different models. We believe that this framework can be used in conjunction with existing approaches to better assess different uncertainty estimation techniques. As we have shown in our toy setting, the study of these predictive distributions under different simulation settings can help form a more complete picture. It is also worth noting that we are also able to obtain insights such as the quality of confidence intervals for classification problems as well. As future work, we will utilize this framework to perform extensive comparisons using more complex simulations with different datasets.

References

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 2015.

Dabney, W., Rowland, M., Bellemare, M. G., and Munos, R. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA, 2016.

Hernández-Lobato, J. M. and Adams, R. P. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Kendall, A. and Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pp. 5574–5584, 2017.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.

McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1222–1230, 2013.

Neal, R. M. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Neklyudov, K., Molchanov, D., Ashukha, A., and Vetrov, D. Variance networks: When expectation does not meet your expectations. *arXiv preprint arXiv:1803.03764*, 2018.

Paass, G. Assessing and improving neural network predictions by the bootstrap algorithm. In *Advances in Neural Information Processing Systems*, pp. 196–203, 1993.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

Romano, Y., Patterson, E., and Candés, E. J. Conformalized quantile regression. In *33rd Conference on Neural Information Processing System*, 2019.

Smith, L. and Gal, Y. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*, 2018.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M. A., and Prabhat, R. P. A. Scalable bayesian optimization using deep neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 2015.

Snoek, J., Ovadia, Y., Fertig, E., Lakshminarayanan, B., Nowozin, S., Sculley, D., Dillon, J., Ren, J., and Nado, Z. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pp. 13969–13980, 2019.

Tagasovska, N. and Lopez-Paz, D. Single-model uncertainties for deep learning. In *Advances in Neural Information Processing Systems*, pp. 6414–6425, 2019.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 1933.