# The Case for Bayesian Deep Learning

Andrew Gordon Wilson
andrewgw@cims.nyu.edu
Courant Institute of Mathematical Sciences
Center for Data Science
New York University

December 30, 2019

## Abstract

Bayesian inference is especially compelling for deep neural networks. The key distinguishing property of a Bayesian approach is marginalization instead of optimization, not the prior, or Bayes rule. Neural networks are typically underspecified by the data, and can represent many different but high performing models corresponding to different settings of parameters, which is exactly when marginalization will make the biggest difference for accuracy and calibration. Moreover, deep ensembles can be seen as approximate Bayesian marginalization.

In many situations, the predictive distribution we want to compute is given by

$$p(y|x, \mathcal{D}) = \int p(y|x, w)p(w|\mathcal{D})dw\,. \tag{1}$$

The outputs are $y$ (e.g., class labels, regression values, ...), indexed by inputs $x$ (e.g. images, spatial locations, ...), the weights (or parameters) of the model $f(x; w)$ are $w$, and $\mathcal{D}$ are the data. Eq. (1) represents a *Bayesian model average* (BMA). Rather than bet everything on one hypothesis — with a single setting of parameters $w$ — we want to use every possible setting of parameters, weighted by their posterior probabilities. This process is called *marginalization* of the parameters $w$, since the predictive distribution of interest no longer conditions on $w$. This is not a controversial equation, but a direct expression of the sum and product rules of probability. The BMA represents *epistemic uncertainty* — that is, uncertainty over which setting of weights (hypothesis) is correct, given limited data. Epistemic uncertainty is sometimes referred to as *model uncertainty*, in contrast to *aleatoric* uncertainty coming from noise in the measurement process. One can naturally visualize epistemic uncertainty in regression, by looking at the spread of the predictive distribution as we move in $x$ space. As we move away from the data, there are many more functions that are consistent with our observations, and so our epistemic uncertainty should grow.

In classical training, one typically finds the *regularized maximum likelihood* solution

$$\hat{w} = \text{argmax}_w \log p(w|\mathcal{D}) = \text{argmax}_w(\log p(\mathcal{D}|w) + \log p(w) + \text{constant}). \tag{2}$$

This procedure is sometimes called *maximum a-posteriori (MAP) optimization*, as it involves maximizing a posterior. $\log p(\mathcal{D}|w)$ is the log likelihood, formed by relating the function

we want to learn $f(x; w)$ to our observations. If we are performing classification with a softmax link function, $-\log p(\mathcal{D}|w)$ corresponds to the cross entropy loss. If we are are performing regression with Gaussian noise, such that $p(\mathcal{D}|w) = \prod_{j=1}^{n} p(y_j|w, x_j) = \prod_{j=1}^{n} \mathcal{N}(y_j; f(x_i; w), \sigma^2)$, then $-\log p(\mathcal{D}|w)$ is a scaled MSE loss. In this context, the prior $p(w)$ acts as a *regularizer*. If we choose a flat prior, which has no preference for any setting of the parameters $w$ (it does not assign any feasible setting any more prior density than any other), then it will have no effect on the optimization solution. On the other hand, a flat prior may have a major effect on *marginalization*. Indeed, even though MAP involves a posterior and a prior, and an instantiation of Bayes rule, it is not at all Bayesian, since it is performing optimization to bet everything on a single hypothesis $f(x; \hat{w})$.

We can view classical training as performing approximate Bayesian inference, using the approximate posterior $p(w|\mathcal{D}) \approx \delta(w = \hat{w})$, where $\delta$ is a Dirac delta function that is zero everywhere except at $\hat{w}$. In this case, we recover the standard predictive distribution $p(y|x, \hat{w})$. From this perspective, many alternatives, albeit imperfect, will be preferable — including impoverished Gaussian posterior approximations for $p(w|\mathcal{D})$, even if the posterior or likelihood are actually highly non-Gaussian and multimodal.

The difference between a classical and Bayesian approach will depend on how sharply peaked the posterior $p(w|\mathcal{D})$ becomes. If the posterior is sharply peaked, there may be almost no difference, since a point mass may then be a reasonable approximation of the posterior. However, deep neural networks are typically very underspecified by the available data, and will thus have diffuse likelihoods $p(\mathcal{D}|w)$. Not only are the likelihoods diffuse, but different settings of the parameters correspond to a diverse variety of compelling explanations for the data. Indeed, Garipov et al. (2018) shows that there are large valleys in the loss landscape of neural networks, over which parameters incur very little loss, but give rise to high performing functions which make meaningfully different predictions on test data. Zołna et al. (2019) also demonstrates the variety of good solutions that can be expressed by a neural network posterior. This is exactly the setting when we *most* want to perform a Bayesian model average, which will lead to an ensemble containing many different but high performing models, for better accuracy and better calibration than classical training.

The recent success of *deep ensembles* (Lakshminarayanan et al., 2017) is not discouraging, but indeed strong motivation for following a Bayesian approach. Deep ensembles involves MAP training of *the same architecture* many times starting from different random initializations, to find different local optima. Thus *using these models in an ensemble is an approximate Bayesian model average*, with weights that correspond to models with high likelihood and diverse predictions. Instead of using a single point mass to approximate our posterior, as with classical training, we are now using multiple point masses in good locations, enabling a better approximation to the integral in Eq. (1) that we are trying to solve. The functional diversity is important for a good approximation to the BMA integral, because we are summing together terms of the form $p(y|x, w)$; if two settings of the weights $w_i$ and $w_j$ each provide high likelihood (and consequently high posterior mass), but give rise to similar models, then they will be largely redundant in the model average, and the second setting of parameters will not contribute much to the integral estimate.

While a recent report (Ovadia et al., 2019) shows that deep ensembles appear to outperform some particular approaches to Bayesian neural networks, there are two key reasons behind these results that are actually optimistic for Bayesian approaches. First, the deep ensembles being used are finding many different basins of attraction, corresponding to diverse solutions, which enables *a better approximation to a Bayesian model average* than the specific Bayesian methods considered in Ovadia et al. (2019), which focus their modelling effort on a *single* basin of attraction. The second is that the deep ensembles require retraining a network from scratch many times, which incurs a great computational expense. If one were to control for computation, the approaches which focus on a single basin may be preferred.

There is an important distinction between a Bayesian model average and some approaches to ensembling. The Bayesian model average assumes that *one* hypothesis (one setting of the weights) is correct, and averages over models due to an inability to distinguish between hypotheses given limited data (Minka, 2000). As we observe more data, the posterior collapses, and the Bayesian model average converges to the maximum likelihood solution. If the true explanation for the data is actually a *combination* of hypotheses, the Bayesian model average may then perform worse as we observe more data. Some ensembling methods instead work by enriching the hypothesis space, and thus do not collapse in this way. Deep ensembles, however, are finding different MAP or maximum likelihood solutions, corresponding to different basins of attraction, starting from different random initializations. Therefore the deep ensemble will collapse when the posterior concentrates, as with a Bayesian model average. Since the hypothesis space is highly expressive for a modern neural network, posterior collapse in many cases is desirable.

Regarding priors, the prior that matters is the prior in *function space*, not parameter space. In the case of a Gaussian process (e.g. Williams and Rasmussen, 2006), a vague prior would be disastrous, as it is a prior directly in function space and would correspond to white noise. However, when we combine a vague prior over parameters $p(w)$ with a structured function form $f(x; w)$ such as a convolutional neural network (CNN), we induce a structured prior distribution over functions $p(f(x; w))$. Indeed, the inductive biases and equivariance constraints in such models is why they work well in classical settings. We can sample from this induced prior over functions by first sampling parameters from $p(w)$ and then conditioning on these parameters in $f(x; w)$ to form a sample from $p(f(x; w))$ (e.g., Wilson, 2014, Ch 2). Alternatively, we can use a neural network kernel with a Gaussian process, to induce a structured distribution over functions (Wilson et al., 2016).

Bayesian or not, the prior, just like the functional form of a model, or the likelihood, will certainly be imperfect, and making unassailable assumptions will be impossible. Attempting to avoid an important part of the modelling process because one has to make assumptions, however, will often be a worse alternative than an imperfect assumption. There are many considerations one might have in selecting a prior. Sometimes a consideration is invariance under reparametrization. Parametrization invariance is also a question in considering regularizers, optimization procedures, model specification, etc., and is not specific to whether or not one should follow a Bayesian approach. Nonetheless, I will make some brief additional remarks on these questions.

If we truly have a vague prior over parameters, perhaps subject to some constraint for normalization, then our posterior reflects essentially the same relative preferences between models as our likelihood, for it is a likelihood scaled by a factor that does not depend on $w$ outside some basic constraints. In computing the integral for a Bayesian model average, each setting of parameters is weighted by the quality of the associated function, as measured by the likelihood. Thus the model average is happening in function space, and is invariant to reparametrization. In the context of many standard architectural specifications, there are also some additional benefits to using relatively broad zero-mean centred Gaussian priors, which help provide smoothness in function space by bounding the norm of the weights. But this smoothness is not a central reason to follow a Bayesian approach, as one could realize similar advantages in performing MAP optimization. Bayesian methods are fundamentally about marginalization as an alternative to optimization.

Moreover, vague priors over parameters are also often a reasonable description of our a priori subjective beliefs. We want to use a given functional form, which is by no means vague, but we often do not have any strong a priori preference for a setting of the parameters. It is worth reiterating that a vague prior in parameter space combined with a highly structured model such as a convolutional neural network does *not* imply a vague prior in function space, which is also why classical training of neural networks provides good results. Indeed, vague parameter priors are often preferable to entirely ignoring epistemic uncertainty, which

would be the standard alternative. In fact, ignoring epistemic uncertainty is a key reason that standard neural network training is *miscalibrated*. By erroneously assuming that the model (parameter setting we want to use) is completely determined by a finite dataset, the predictive distribution becomes *overconfident*: for example, the highest softmax output of a CNN that has undergone standard training (e.g. MAP optimization) will typically be much higher than the probability of the corresponding class label (Guo et al., 2017). Importantly, ignoring epistemic uncertainty also leads to worse accuracy in point predictions, because we are now ignoring all the other compelling explanations for the data. While improvements in calibration are an empirically recognized benefit of a Bayesian approach, the enormous potential for gains in *accuracy* through Bayesian marginalization with neural networks is a largely overlooked advantage.

There are also many examples where flat priors over parameters combined with *marginalization* sidestep pathologies of maximum likelihood training. Priors without marginalization are simply regularization, but Bayesian methods are not about regularization (MacKay, 2003, Ch 28). And there is a large body of work considering approximate Bayesian methods with uninformative priors over parameters (*but not functions*) (e.g., Clyde and George, 2004; Berger and Pericchi, 1996; O'Hagan, 1995; Berger et al., 2006; Gelman et al., 2013; MacKay, 2003, 1992a; Neal, 1996). This approach is well-motivated, marginalization is still compelling, and the results are often better than regularized optimization.

By accounting for epistemic uncertainty through uninformative *parameter* (but not function) priors, we, as a community, have developed Bayesian deep learning methods with improved calibration, reliable predictive distributions, and improved accuracy (e.g., MacKay, 1992b; Neal, 1996; Gal and Ghahramani, 2016; Saatci and Wilson, 2017; Kendall and Gal, 2017; Ritter et al., 2018; Khan et al., 2018; Maddox et al., 2019; Sun et al., 2019; Izmailov et al., 2019; Zhang et al., 2020). MacKay (1995) and Neal (1996) are particularly notable early works considering Bayesian inference for neural networks. Seeger (2006) also provides a clear tutorial on Bayesian methods in machine learning. Of course, we can always make better assumptions — Bayesian or not. We should strive to build more interpretable parameter priors. There are works that consider building more informative parameter priors for neural networks by reasoning in function space (e.g., Sun et al., 2019; Yang et al., 2019; Louizos et al., 2019; Hafner et al., 2018). And we should also build better posterior approximations. Deep ensembles are a promising step in this direction.

But we should not undermine the progress we are making so far. Bayesian inference is especially compelling for deep neural networks. Bayesian deep learning is gaining visibility because we are making progress, with good and increasingly scalable practical results. We should not discourage these efforts. If we are shying away from an approximate Bayesian approach because of some challenge or imperfection, we should always ask, *"what is the alternative"?* The alternative may indeed be a more impoverished representation of the predictive distribution we want to compute.

There are certainly many challenges to computing the integral of Eq. (1) for modern neural networks, including a posterior landscape which is difficult to navigate, and an enormously high (e.g., 30 million) dimensional parameter space. Many of the above papers are working towards addressing such challenges. We have been particularly working on recycling geometric information in the SGD trajectory for scalable approximate Bayesian inference (Izmailov et al., 2019; Maddox et al., 2019), exploiting large loss valleys (Garipov et al., 2018), and creating subspaces of low dimensionality that capture much of the variability of the network (Izmailov et al., 2019). Pradier et al. (2018) also considers different approaches to dimensionality reduction, based on non-linear transformations. For exploring multiple distinct basins of attraction, we have been developing cyclical stochastic MCMC approaches (Zhang et al., 2020), which could be seen as sharing many of the advantages of deep ensembles, but with an added attempt to also marginalize within basins of attraction.

# References


Berger, J. et al. (2006). The case for objective Bayesian analysis. *Bayesian analysis*, 1(3):385–402.

Berger, J. O. and Pericchi, L. R. (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122.

Clyde, M. and George, E. I. (2004). Model uncertainty. *Statistical science*, pages 81–94.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. (2018). Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Neural Information Processing Systems*.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org.

Hafner, D., Tran, D., Irpan, A., Lillicrap, T., and Davidson, J. (2018). Reliable uncertainty estimates in deep neural networks using noise contrastive priors. *arXiv preprint arXiv:1807.09289*.

Izmailov, P., Maddox, W. J., Kirichenko, P., Garipov, T., Vetrov, D., and Wilson, A. G. (2019). Subspace inference for Bayesian deep learning. In *Uncertainty in Artificial Intelligence*.

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

Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018). Fast and scalable bayesian deep learning by weight-perturbation in adam. *arXiv preprint arXiv:1806.04854*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413.

Louizos, C., Shi, X., Schutte, K., and Welling, M. (2019). The functional neural process. In *Advances in Neural Information Processing Systems*.

MacKay, D. J. (1992a). Bayesian interpolation. *Neural Computation*, 4(3):415–447.

MacKay, D. J. (1992b). *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology.

MacKay, D. J. (1995). Probable networks and plausible predictions?a review of practical bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469–505.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Maddox, W., Garipov, T., Izmailov, P., Vetrov, D., and Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*.

Minka, T. P. (2000). Bayesian model averaging is not model combination. *Available electronically at http://www.stat.cmu.edu/minka/papers/bma.html*.

Neal, R. (1996). *Bayesian Learning for Neural Networks*. Springer Verlag.

O'Hagan, A. (1995). Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):99–118.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*.

Pradier, M. F., Pan, W., Yao, J., Ghosh, S., and Doshi-Velez, F. (2018). Latent projection bnns: Avoiding weight-space pathologies by learning latent representations of neural network weights. *arXiv preprint arXiv:1811.07006*.

Ritter, H., Botev, A., and Barber, D. (2018). A scalable Laplace approximation for neural networks. In *International Conference on Learning Representations (ICLR)*.

Saatci, Y. and Wilson, A. G. (2017). Bayesian GAN. In *Advances in neural information processing systems*, pages 3622–3631.

Seeger, M. (2006). Bayesian modelling in machine learning: A tutorial review. Technical report.

Sun, S., Zhang, G., Shi, J., and Grosse, R. (2019). Functional variational bayesian neural networks. *arXiv preprint arXiv:1903.05779*.

Williams, C. K. and Rasmussen, C. E. (2006). Gaussian processes for machine learning. *the MIT Press*, 2(3):4.

Wilson, A. G. (2014). *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, University of Cambridge.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2016). Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378.

Yang, W., Lorch, L., Graule, M. A., Srinivasan, S., Suresh, A., Yao, J., Pradier, M. F., and Doshi-Velez, F. (2019). Output-constrained bayesian neural networks. *arXiv preprint arXiv:1905.06287*.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. (2020). Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*.

Zołna, K., Geras, K. J., and Cho, K. (2019). Classifier-agnostic saliency map extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 10087–10088.