Imposing Functional Priors on Bayesian Neural Networks

Bogdan Kozyrskiy, Dimitrios Milios, Maurizio Filippone

Imposing Functional Priors on Bayesian Neural Networks

Bogdan Kozyrskiy[1], Dimitrios Milios[2], Maurizio Filippone[1]
[1]*Department of Data Science, EURECOM, 450 Route des Chappes, Biot, France*
[2]*Jubile Tech Ltd., London, UK*
{*Bogdan.Kozyrskiy, Maurizio.Filippone*}*@eurecom.fr, dimitrios.milios@gmail.com*

Keywords: Bayesian Inference, Markov Chain Monte-Carlo, Deep Neural Networks

Abstract: Specifying sensible priors for Bayesian neural networks (BNNs) is key to obtain state-of-the-art predictive performance while obtaining sound predictive uncertainties. However, this is generally difficult because of the complex way prior distributions induce distributions over the functions that BNNs can represent. Switching the focus from the prior over the weights to such functional priors allows for the reasoning on what meaningful prior information should be incorporated. We propose to enforce such meaningful functional priors through Gaussian processes (GPs), which we view as a form of implicit prior over the weights, and we employ scalable Markov chain Monte Carlo (MCMC) to obtain samples from an approximation to the posterior distribution over BNN weights. Unlike previous approaches, our proposal does not require the modification of the original BNN model, it does not require any expensive preliminary optimization, and it can use any inference techniques and any functional prior that can be expressed in closed form. We illustrate the effectiveness of our approach with an extensive experimental campaign.

1 INTRODUCTION

Artificial Neural Networks (NN) currently represent a general class of successful models for various machine learning tasks, including computer vision, natural language processing, and many others. Bayesian Neural Networks (BNN) combine the representation power of NNs with Bayesian inference, making them an attractive choice in applications where predictive performance and accurate uncertainty quantification is important. BNNs are difficult to use because of the intractability of the posterior over model parameters, which necessitates approximations. Choosing appropriate priors over model parameters is also crucial for good performance (Fortuin, 2022; Tran et al., 2022). In BNNs, the prior over the weights and the network architecture determine a distribution over the outputs of such BNNs (Sun et al., 2019), and we refer to this induced prior as a *functional prior*. The functional prior should encode any prior information on the conditional distribution of the labels given the inputs. However, it is unclear how to encode this type of information when having to specify a prior distribution over the weights.

This paper presents a framework for imposing meaningful functional priors using scalable Markov chain Monte Carlo (MCMC) sampling from an approximation to the posterior distribution over BNN weights, and we specify the prior over the weights implicitly through a prior over the induced functional prior. Our approach is different from the literature on Implicit Process Priors (IPPs) (Ma et al., 2019), where the goal is to obtain an approximate framework to handle the functional prior implicitly induced by the choice of a prior distribution over the weights. In our work, we operate in the opposite direction by *imposing* a functional prior, which implicitly determines a prior over the weights; we do not know such a prior over the weights in closed form, but we implicitly determine it through the specification of the induced functional prior.

Stochastic Processes are natural mathematical objects suitable to define distributions over functions (Kallenberg and Kallenberg, 1997), and Gaussian Processes (GPs) represent popular examples which are routinely used in numerous machine learning tasks. This type of stochastic processes is well investigated and has strong theoretical foundations (Williams and Rasmussen, 2006). There are theoretical guarantees for the generalization error of GP regression, and this method has a strong connection with non-Bayesian Kernel Ridge Regression (KRRs) (Kanagawa et al., 2018). Also, it was shown in (Neal, 1996) that in the infinite width limit, shallow BNNs are equivalent to GPs. We propose to use GPs to impose functional priors over BNNs because GPs pro-

vide a flexible set of tools to encode different types of beliefs about functions, such as periodicity or smoothness through the specification of kernels. However, our approach is not restricted to GPs, and it can handle any functional priors that can be written down in closed form.

This paper is organized as follows. We review the related literature in Sec. 2, and we present our method in Sec. 3. We report results on various benchmarks in Sec. 4 and we conclude the paper in Sec. 6, after discussing the limitations of our work in Sec. 5.

2 RELATED WORK

A popular way of choosing prior distributions for BNNs is to employ a Gaussian distribution over the weight of the model (Graves, 2011; Neal, 1996). This offers some practical advantages, for instance when employing Variational Inference (VI) (Graves, 2011). Mean-field VI allows for efficient calculations of the regularization part of the VI objective function without the need to resort to Monte Carlo approximations, but the limited flexibility of the approximating distributions may negatively affect performance.

Even when adopting more advanced and generally more accurate inference techniques, such as Stochastic Gradient Hamiltonian Monte Carlo (SG-HMC) (Chen et al., 2014), the Gaussian assumption on the prior over model parameters is still common. It was shown by (Fortuin et al., 2021) that Gaussian priors are problematic in terms of model performance and the ability to detect Out-of-Domain (OOD) input examples. This work also shows how Gaussian priors over the weights could be responsible for the cold posterior effect described by (Wenzel et al., 2020); this effect is characterized by the necessity of applying temperature scaling to the prior density term in Bayes theorem in order to obtain good performance.

Flexible alternatives to Gaussian priors, such as mixture of Gaussians (Blundell et al., 2015), Student's t-distribution (Fortuin et al., 2021), hierarchical Gaussian distribution (Chen et al., 2014) and many others (Fortuin, 2022) were developed to address poor performance of Gaussian priors. However, all these types of priors still do not help understanding their effect on model outputs.

An alternative to studying weight priors is to focus on their effect on NNs functional priors. A variational objective computed on a finite set of function evaluations is proposed in (Sun et al., 2019) for finding a Bayesian posterior in the space of functions for a functional prior defined by a stochastic process. The authors show that the supremum of the KL divergence over all sets of input points is equal to the true KL divergence in functional space. In this setting, the optimization procedure simultaneously minimizes the optimization objective with respect to the parameters of the model and maximizes the KL term with respect to the input data points, which makes the optimization process unstable. Also, the optimization objective requires evaluating the gradient of the approximate posterior density by the Stein gradient estimator (Shi et al., 2018), and this requires a careful choice of a kernel function. The work in (Ma et al., 2019) focuses on representing the functional prior as a BNN and uses GPs to obtain an approximate posterior over functions. The problem with this approach is that GPs may yield a poor approximation quality for the true functional posterior. The authors in (Sun et al., 2019) and (Ma et al., 2019) use VI to find an approximate posterior distribution, which means that the optimization objective contains a functional KL divergence term. However, in (Rudner et al., 2021) it is claimed that the KL divergence between the functional approximate posterior and the GP process functional prior is problematic as it may diverge to infinity. On the other hand, they acknowledge that it does not mean that parametric models cannot approximate GPs well.

The authors of (Tran et al., 2022) propose to impose functional GP priors so as to constrain the parametric prior over the weights of BNNs. They propose to optimize parameters of the prior over the weights by minimizing the Wasserstein distance between the BNN functional prior and the GP prior. Then, the posterior over the weights is characterized by means of MCMC.

In our work, we aim to avoid the computation of the KL divergence or any other distance metric in function spaces. Instead, we propose to enforce the choice of a functional prior directly when carrying out approximate inference of BNN weights.

3 METHODS

Consider a supervised learning task with a dataset $\mathcal{D}\{(\mathbf{x}_i, y_i)\}_{i=1\ldots n}$ of n input vectors $\mathbf{X} = \{\mathbf{x}_i\}_{i=1\ldots,n}$ and corresponding labels $\mathbf{y} = \{y_i\}_{i=1\ldots,n}$, and imagine employing a NN-based model with parameters $\mathbf{w}$ to establish a parametric mapping between inputs and labels. We denote the input/output mapping by $f_{\mathbf{w}}(\mathbf{x})$, and for convenience we also define $\mathbf{f}^\top = [f_{\mathbf{w}}(\mathbf{x}_1), \ldots, f_{\mathbf{w}}(\mathbf{x}_N)]$ and $\mathbf{f}^{*\top} = [f_{\mathbf{w}}(\mathbf{x}_1), \ldots, f_{\mathbf{w}}(\mathbf{x}_N), f_{\mathbf{w}}(\tilde{\mathbf{x}}_1), \ldots, f_{\mathbf{w}}(\tilde{\mathbf{x}}_M)]$ as the evaluation of the function $f_{\mathbf{w}}(\mathbf{x})$ at the inputs $\mathbf{X}$ and an augmented set of inputs $\mathbf{X}^* = [\mathbf{X}, \tilde{\mathbf{X}}]$, respectively. The set

$\mathbf{X}^*$ has cardinality $N^* = N + \tilde{N}$, and the $\tilde{N}$ inputs in $\tilde{\mathbf{X}}$ are drawn from a given $p(\mathbf{x})$. Note that the sets $\mathbf{X}$ and $\mathbf{X}^*$ can be disjoint, but in order to keep the notation uncluttered, we assume $\mathbf{X} \subset \mathbf{X}^*$

3.1 Imposing Functional Priors on BNNs

A Bayesian treatment NNs requires specifying a prior distribution $p(\mathbf{w})$ over the parameters and a likelihood function for the labels given the inputs, that is $p(\mathbf{y}|\mathbf{X}, \mathbf{w})$. For this BNN, it is possible to write down an expression for the posterior distribution over model parameters as:

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})d\mathbf{w}} \tag{1}$$

Carrying out inference in BNNs is extremely difficult for at least two reasons. One main difficulty stems from the complex way in which parameters affect the likelihood function, and this requires approximation techniques to characterize the posterior over model parameters; popular approaches involve MCMC and variational approximations. A second and more subtle challenge is how to specify priors for BNNs, because it is difficult to establish what is the effect of prior parameters on the distribution over the functions that BNNs can represent. In this work, we propose a novel way to address the challenge of choosing sensible priors for BNNs by working with implicit priors over the weights induced by the choice of functional priors, while we follow the recent trend to employ MCMC techniques to address the intractability of the inference process. We begin by focusing on the distribution over the functions represented by BNNs. In particular, we consider the distribution of $\mathbf{f}^*$, which is the distribution of $f_{\mathbf{w}}(\mathbf{x})$ evaluated at the set of input points $\mathbf{X}^*$, and we impose a prior over this set of variables which encourages functions to behave in a sensible way *a priori*. Later we will study in particular Gaussian process priors, but any functional prior can be incorporated as long as it can be expressed in closed form.

We now rewrite the likelihood function in terms of $\mathbf{f}$ rather than $\mathbf{w}$:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) \rightarrow p(\mathbf{y}|\mathbf{f}). \tag{2}$$

The main idea behind our work is to now define a prior over $\mathbf{f}$ instead of $\mathbf{w}$, and to perform inference over $\mathbf{w}$. With this change of variables, we should account for the change of measure through a Jacobian term. However, such a change of variables involves groups of variables of different dimensions in general and even when this is not the case, computing this term would be computationally costly. For this reason, we are going to ignore the Jacobian accepting to settle for an approximate posterior over $\mathbf{w}$. With this choice, we rewrite Bayes theorem as:

$$\log p(\mathbf{f}^*|\mathbf{y}, \mathbf{X}^*) = \log p(\mathbf{y}|\mathbf{f}) + \log p(\mathbf{f}^*|\mathbf{X}^*) + \text{const.} \tag{3}$$

Note that in this equation we introduced the functional prior:

$$p(\mathbf{f}^*|\mathbf{X}^*) = \int p(\mathbf{f}^*|\mathbf{X}^*, \mathbf{w})p(\mathbf{w})d\mathbf{w}, \tag{4}$$

where $p(\mathbf{f}^*|\mathbf{X}^*, \mathbf{w})$ is a Dirac's delta placed at the evaluation of $f_{\mathbf{w}}(\mathbf{x})$ at the inputs $\mathbf{X}^*$ due to the deterministic way in which inputs are mapped into outputs in NNs. Again, we stress that while we focus on the distribution of functions represented by BNNs, we actually use the objective in eq. 3 to perform MCMC sampling in the space of the weights $\mathbf{w}$. Note that we carry out inference over $\mathbf{w}$ through MCMC, but given that we are working with an approximation to the posterior over $\mathbf{w}$, we could alternatively employ other fast approximate inference techniques such as VI. Here, we focus on MCMC so as to isolate the effect of the way we impose functional priors compared with alternatives which try to characterize the exact posterior over $\mathbf{w}$ (Tran et al., 2022).

Bayesian interpretation. From a Bayesian point of view, imposing a prior over function by specifying a prior over $\mathbf{f}^*$ induces an implicit prior over the weights through eq. 4. In other words, the prior over $\mathbf{f}^*$ is in practice a prior over a deterministic transformation of $\mathbf{w}$, and this is implemented by the NN. It is interesting to note that in the literature eq. 4 is usually interpreted in the opposite way; that is, one uses eq. 4 starting from a prior over the weights $p(\mathbf{w})$ to define a functional prior in an implicit way (Ma et al., 2019). The likelihood function establishes what is the likelihood of the labels $\mathbf{y}$ and it is conditioned on $\mathbf{f}$ or equivalently on $\mathbf{w}$ and $\mathbf{X}$. Therefore, the expression in eq. 3 can be seen as an expression for the (approximate) posterior over the weights $\mathbf{w}$ (due to the lack of a Jacobian term), where the prior is assumed over a transformation of such weights. In this paper, we take this view to carry out Bayesian inference over $\mathbf{w}$ using MCMC techniques. We also note that our approach has some close similarity with the Product of Expert approach proposed in (Wenk et al., 2019) for inference of parameters of Ordinary Differential Equations using Gaussian Processes.

Regularization interpretation. While we proceed with a Bayesian treatment of $\mathbf{w}$, it is useful to interpret eq. 3 as a regularized objective in the following

way. The first term $\log p(\mathbf{y}|\mathbf{f})$ is the negative loss, which can be equivalently be seen as a function of $\mathbf{w}$ and $\mathbf{X}^*$, so this provides a constraint on $\mathbf{w}$ because the objective promotes values of $\mathbf{f}$ which are compatible with the labels $\mathbf{y}$, and $\mathbf{f}$ depends on $\mathbf{w}$ and $\mathbf{X}^*$. The second term is a regularization term, which penalizes functions deviating from a behavior established by the functional prior. Because $\mathbf{f}^*$ is a function of $\mathbf{w}$ and $\mathbf{X}^*$, this translates into a regularization term for $\mathbf{w}$.

3.2 Imposing Functional Priors through Gaussian Processes

The proposed formulation focusing on functional representations has the advantage of putting the emphasis on the functions that BNNs can represent, and for which it is possible to assume sensible priors. Here we specify how to operate in case of Gaussian processes (GPs), which yield a prior term in eq. 3 as:

$$\log p(\mathbf{f}^*|\mathbf{X}^*) = -\frac{1}{2}\mathbf{f}^{*\top}\mathbf{C}^{-1}\mathbf{f}^* + \text{const}, \qquad (5)$$

where the covariance matrix is $\mathbf{C} = (\mathbf{K}_{\mathbf{X}^*\mathbf{X}^*} + \sigma_n^2)$, and $\mathbf{K}_{\mathbf{X}^*\mathbf{X}^*}$ contains the evaluation of the kernel function κ among all the inputs in $\mathbf{X}^*$. For simplicity, we assume a zero-mean GP, but other mean functions can be easily included. In the next subsections, we elaborate on how to use this GP prior in practice, by proposing a way to operate with mini-batches for scalability purposes, by discussing hyper-parameter optimization, and by discussing the properties of the proposed approach when N^* goes to infinity.

3.2.1 Mini-batching

In this work, we aim to employ advanced MCMC sampling methods based on stochastic gradients, and in particular Stochastic Gradient Hamiltonian Monte Carlo (SG-HMC) (Chen et al., 2014) to sample from the weights $\mathbf{w}$ of BNNs. In order to do so, we need to formulate our MCMC objective in a way that is suitable for mini-batching. However, extending the previous formulation to operate with mini-batches without care would produce a biased estimation of the quadratic term $\mathbf{f}^{*\top}\mathbf{C}^{-1}\mathbf{f}^* \neq \mathbb{E}[\mathbf{f}_b^{*\top}\mathbf{C}_b^{-1}\mathbf{f}_b^*]$, where $\mathbf{f}_b$ and $\mathbf{C}_b$ are computed over a mini-batch $\mathbf{X}_b$.

The main difficulty of full batch training is the necessity of solving linear systems with the matrix $\mathbf{C}$, which has $O(N^{*3})$ complexity in the number of inputs in $\mathbf{X}^*$. The literature on GPs offers many cues on how to circumvent this problem. In particular, there exist formulations of GPs based on random features (Rahimi and Recht, 2007) which operate on mini-batches (Cutajar et al., 2017). In this work, we focus on approximations based on random features, but inducing points formulations are also possible.

Random Feature (RF) expansions of the kernel $\kappa(\cdot,\cdot)$ allow one to obtain a finite-dimensional representation for an explicit feature map which approximates the true possibly infinite-dimensional feature map. Using this expansion, we can express the Gram matrix as a dot product of feature maps computed over the data $\mathbf{K} \approx \Phi\Phi^\top$. We can use this property and the Woodbury identity to rewrite the quadratic term as follows:

$$\mathbf{f}^{*\top}\mathbf{C}^{-1}\mathbf{f}^* = \mathbf{f}^{*\top}(\Phi\Phi^\top + \sigma_f^2\mathbf{I})^{-1}\mathbf{f}^* = \frac{1}{\sigma_f^2}\mathbf{f}^{*\top}\mathbf{f}^* - \frac{1}{\sigma_f^2}\mathbf{f}^{*\top}\Phi(\Phi^\top\Phi + \sigma_f^2\mathbf{I})^{-1}\Phi^\top\mathbf{f}^*. \qquad (6)$$

In this case, instead of inverting a matrix of size $N^* \times N^*$, we invert a matrix of size $D \times D$, where D is the dimensionality of the RF vector. However, this approach has two drawbacks. First, it is unstable when $\sigma_f^2 \to 0$, because after the application of the Woodbury identity the term $\frac{1}{\sigma_f^2}\mathbf{f}^{*\top}\mathbf{f}^* \to \infty$. Second, this approach still does not allow mini-batching.

We can reformulate our MCMC objective by replacing the nonparametric term pertaining to the GP with a parametric one based on RFs. For the set $\mathbf{f}^*$, we can factorize its prior probability as:

$$p(\mathbf{f}^*|\mathbf{X}^*) = \int p(\mathbf{f}^*|\beta, \mathbf{X}^*)p(\beta)d\beta, \qquad (7)$$

where β are the parameters of RF approximation of the GP, that is $p(\beta) \sim \mathcal{N}(0, \mathbf{I})$ and $p(\mathbf{f}^*|\beta, \mathbf{X}^*) \sim \mathcal{N}(\Phi\beta, \sigma_f^2\mathbf{I})$. In this case it is easy to verify that $p(\mathbf{f}^*) = \mathcal{N}(0, \Phi\Phi^\top + \sigma_f^2\mathbf{I})$ and according to the property of the RF approximation, the covariance matrix coincides with the prior term of the objective in eq. 6. Instead of sampling directly from the unnormalized posterior $p(\mathbf{f}^*|\mathbf{X}^*, \mathbf{y})$ marginalized over β, we can sample from the joint density $p(\mathbf{f}^*, \beta|\mathbf{X}, \mathbf{y})$ and discard samples over β:

$$p(\mathbf{f}^*, \beta|\mathbf{X}^*, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{f})p(\mathbf{f}^*|\beta, \mathbf{X}^*)p(\beta). \qquad (8)$$

Again, when we refer to the fact that we sample $\mathbf{f}^*$, in practice we sample $\mathbf{w}$. This RF-based approach avoids the necessity of inverting the matrix $(\Phi\Phi^\top + \sigma_f^2\mathbf{I})$ during the computation of the objective.

Resuming, the expression for the unnormalized log-posterior in eq. 8, where the GP regularization is approximated using RFs, is as follows:

$$\log p(\mathbf{y}|\mathbf{f}) - \frac{1}{2\sigma_f^2}\|\mathbf{f}^* - \Phi\beta\|^2 - \frac{||\beta||^2}{2} + \text{const}. \qquad (9)$$

It is straightforward to verify that this MCMC objective can be written as a sum of terms involving

individual input points, and it is therefore amenable to mini-batching. It is also easy to verify that one can proceed with a Gibbs sampling scheme whereby $\mathbf{f}^*$ (that is $\mathbf{w}$) is sampled from the conditional $\hat{p}(\mathbf{f}^*|\beta, \mathbf{X}^*, \mathbf{y})$ using SG-HMC and β is sampled directly from $\hat{p}(\beta|\mathbf{f}^*, \mathbf{X}^*, \mathbf{y})$, which has a Gaussian form.

3.2.2 Hyper-parameter optimization

The choice of a GP prior opens to the need to specify its kernel parameters. In the absence of any way to determine such hyper-parameters, we propose to optimize them by marginal log-likelihood (MLL) optimization, which is a popular way to proceed with GP models. In our case, the random feature approximation lends itself to a scalable solution, avoiding the need to invert large matrices. Again, using Woodbury matrix identities, it is possible to rewrite the marginal likelihood so that the cost of computing it is cubic in the number of random features instead of cubic in the number of input points.

3.2.3 Classification

While for regression it is natural to specify functional priors through GPs and to obtain a tractable framework to scale these through random features, for other likelihoods things may become more involved. For instance, in classification problems, we may wish to specify functional priors such that the distribution over classes is uniform *a priori*.

Alternatively, following an empirical Bayes approach, we could optimize the GP prior hyper-parameters so as to maximize the marginal likelihood. In this case, the random feature approximation of GPs leads to so-called Generalized Linear Models (GLMs) and this requires approximations to be able to compute the marginal likelihood. For classification tasks, there exist solutions to bypass the need to work directly with Bernoulli or Multinoulli likelihoods $p(\mathbf{y}|\mathbf{w}, \mathbf{X})$. Here we follow the idea proposed by (Milios et al., 2018), in which labels are transformed so that classification models can be replaced by regression models with heteroskedastic observation noise. In particular, for each one-hot encoded label $\mathbf{y}$ we can obtain real valued vectors $\tilde{\mathbf{y}}, \sigma_n^2$ (see (Milios et al., 2018) for details):

$$\tilde{y}_i = \log(\alpha_i) - \frac{\tilde{\sigma}_i^2}{2}; \qquad \tilde{\sigma}_i^2 = \log\left(\frac{1}{\alpha_i} + 1\right). \quad (10)$$

With this transformation, we can use a Gaussian likelihood which is conjugate to the Gaussian prior, and thus we can obtain a closed form solution for the marginal likelihood of the model.

4 EXPERIMENTS

4.1 Toy regression dataset

We test our approach on a 1D synthetic dataset using a two-hidden layer NN with *tanh* activation and 256 neurons per layer. The functional GP prior uses an RBF kernel with length-scale $l = 1$ and output variance $\sigma_{out}^2 = 1$. Fig. 1 shows functions sampled from the predictive posterior of the BNN with this GP prior (GP in the figure) as well as the same GP prior approximated with 100 random features with and without mini-batching (GP RFF and GP RFF mini-batch in the figure). We also include the approach from (Tran et al., 2022) which optimizes the Wasserstein distance between the BNN functional prior and the GP prior to determine the prior over BNN weights (WDGPi-G in the figure). For the models with functional prior we used a regularization set of 200 equally spaced test points.

4.2 UCI regression datasets

We tested our approach on UCI datasets (Dua and Graff, 2017) using a two-hidden layer MLP with *tanh* activation and 100 neurons per layer, except for the Protein dataset for which we used 200 neurons. We imposed a GP prior with an RBF kernel and standardized the input vectors and labels. We used the extended dataset $\mathbf{X}^*$, which consists of 90% training data and 10% of uniformly sampled vectors from the input domain, for all experiments. Full-batch training was used for all datasets, while mini-batch training with a batch size of 512 was used for Kin8nm, Power, and Protein.

As a baseline, we consider the aforementioned WDGPi-G method with a Gaussian prior over weights and a Hierarchical GP with a LogNormal distribution over the GP kernel length-scale and output variance. We compare our method to WDGPi-G and deep ensembles (Lakshminarayanan et al., 2017) in terms of RMSE, as shown in Table 1. Each model in the ensemble had the same architecture as the NN in our method.

According to the results, our method is competitive with WDGPi-G on most datasets. It is worthy to note that WDGPi-G uses a Hierarchical Gaussian Process as a functional prior, while our method uses a simple GP. Hierarchical GPs represent a richer functional prior, but we still achieve competitive performance.

We tested the proposed method on the Power dataset with deeper NN architectures featuring four and six layers, and compared its RMSE with Deep

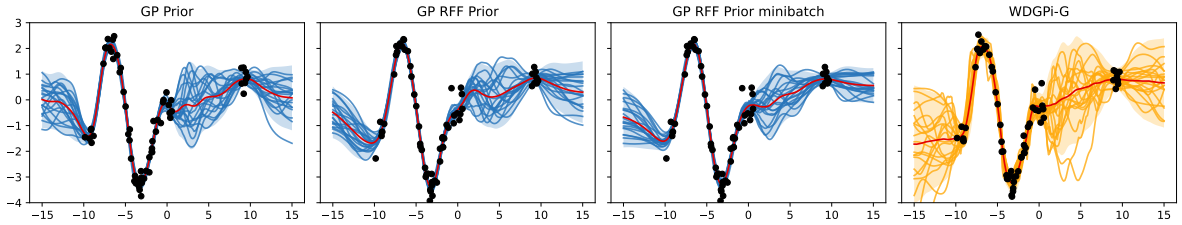

Figure 1: Sampled predictions of BNNs where the GP functional prior is imposed implicitly (our work) and by means of the optimization of the Wasserstein distance with the functional BNN prior (WDGPi-G).

Table 1: Average RMSE for UCI regression datasets

Dataset	Functional MCMC	WDGPi-G	Deep Ensembles
Boston	2.73±0.02	2.83±0.92	3.69±1.15
Concrete	4.06±0.12	4.80±0.41	5.22±0.63
Energy	0.48±0.18	0.34±0.07	1.37±0.32
Kin8nm	0.04±0.00	0.06±0.00	0.06±0.00
Power	3.24±0.06	3.72±0.18	3.86±0.21
Protein	3.61 ±0.04	3.65±0.02	4.45±0.02
Wine	0.60±0.01	0.60±0.04	0.62±0.02

Table 2: MNLL for UCI regression datasets

Dataset	Functional MCMC	WDGPi-G	Deep Ensembles
Boston	2.45±0.01	2.48±0.12	3.19±1.12
Concrete	2.74±0.16	3.03±0.05	3.07±0.26
Energy	0.80±0.05	0.35±0.15	2.07±0.98
Kin8nm	-1.46±0.11	-1.23±0.01	-1.32±0.08
Power	2.73±0.08	2.74±0.04	2.74±0.05
Protein	2.73±0.01	2.75±0.00	2.80±0.01
Wine	0.76±0.04	0.92±0.06	1.08±0.20

Ensembles over iterations (Fig. 2).

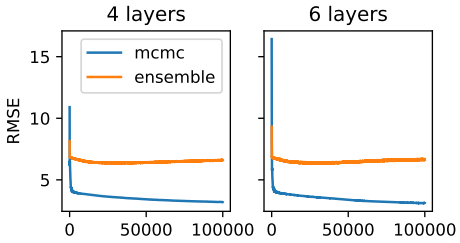

Figure 2: Convergence of RMSE on test data for the Power dataset

4.3 Toy classification dataset

We demonstrate the proposed approach on a 2D toy example using the banana dataset and a two-hidden layer NN with *tanh* activation and 256 neurons per layer. We transform the labels using the method from (Milios et al., 2018) to allow for a Gaussian likelihood, as described in Sec. 3. We use an RBF kernel with $\sigma_{\text{out}} = 5$ and varying length-scales, and compare to the WDGPi-G method from (Tran et al., 2022). We use a grid of 40×40 points as a regularization set and test set. The plot shows that WDGPi-G fails to incorporate the GP prior for a small length-scale ($l = 0.1$) and the prediction function is smoother than expected.

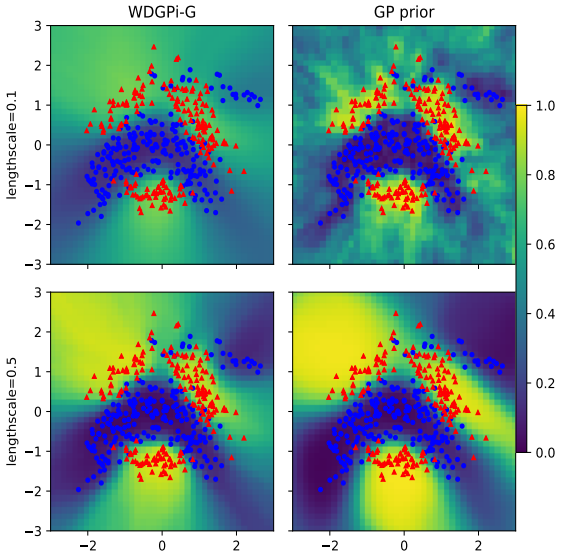

Figure 3: Sampled predictions of the neural network with GP prior using Mahalanobis regularization and WDGPi-G methods

4.4 UCI classification datasets

In this section, we test our approach on various UCI classification datasets using a two-hidden layer NN with *tanh* activation. We use 100 neurons in each hidden layer for EEG, HTRU2, Letter, and Magic, and 200 neurons for Miniboo, Drive, and Mocap. We use the RF approximation of the functional GP prior with $D = 1000$ random features and mini-batches of size 512 on all datasets. GP hyper-parameters are optimized using the label transformation from Sec. 3. We

Table 3: Average classification accuracy for UCI classification datasets

Dataset	Functional MCMC	WDGPi-G	Deep Ensembles
EEG	92.51±1.82	94.13±1.96	89.04 ± 5.01
HTRU2	98.10±0.26	98.03±0.24	98.03 ± 0.20
Magic	88.16±0.33	88.37±0.29	87.90 ± 0.24
Miniboo	92.54±0.21	92.74±0.39	91.49 ± 0.19
Letter	98.22±0.18	96.90±0.29	96.38 ± 0.30
Drive	99.45±0.09	99.69±0.04	99.33 ± 0.05
Mocap	99.10±0.12	99.24±0.10	99.10 ± 0.08

Table 4: Average test NLL for UCI classification datasets

Dataset	Functional MCMC	WDGPi-G	Deep Ensembles
EEG	0.33±0.04	0.18±0.04	0.24 ± 0.10
HTRU2	0.06±0.002	0.06±0.00	0.07 ± 0.01
Magic	0.31±0.00	0.29±0.00	0.30 ± 0.01
Miniboo	0.18±0.01	0.18±0.00	0.20 ± 0.01
Letter	0.09±0.01	0.17±0.00	0.15 ± 0.01
Drive	0.08±0.01	0.03±0.00	0.05 ± 0.01
Mocap	0.19±0.00	0.03±0.00	0.04 ± 0.00

found that using this transformation with the BNN itself gave slightly better results than using classification likelihoods, so we report these results in the table. We attribute this to the optimization of GP hyperparameters with the transformed labels.

We compare our approach with other classification methods and found that it performs competitively with the state-of-the-art, as shown in Tables 3 and 4. Our approach does not require the Wasserstein optimization phase used in WDGPi-G, while still achieving similar classification performance after optimizing GP hyperparameters.

We also tested the proposed method on the Letter dataset using NNs with four and six hidden layers and compared its convergence to the Deep Ensemble approach in terms of classification accuracy (Fig. 4).

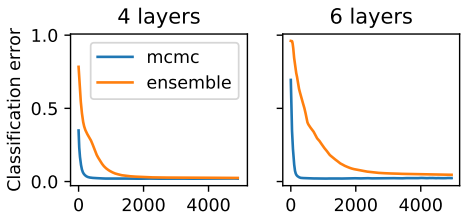

Figure 4: Convergence of classification error on test data for the Letter dataset

5 Limitations

While we consider our approach quite elegant in encoding prior information in the form of functional priors, we believe that it is important to point out some limitations compared to other works.

One limitation is that the posterior distribution we are targeting is approximate due the way we treat the change of variables from weights to functions.

Another limitation is that the functional prior needs to have a closed form. Even though the class of functional priors which have this property is large, this might be too restrictive in applications where it is possible to sample from such priors but no closed form is available. Prior works which perform a preliminary optimization of the prior over the weights (e.g., (Tran et al., 2022)) can operate on samples from functional priors without the need to express these in closed form.

Finally, the choice of a GP prior requires setting its hyper-parameters. In this work, we resort to marginal likelihood optimization, but it is possible that this choice induces overfitting. One way around this would be to include hyper-parameters in the set of variables to be sampled in SG-HMC to obtain samples from their posterior at the expenses of having to deal with a more costly MCMC sampling. Having said that, there are situations where functional priors are easy to elicit and express without the need to carry out hyper-parameter optimization.

6 Conclusions

In this paper, we proposed a novel way to incorporate prior knowledge in Bayesian NNs (BNNs) in the form of functional priors. In our view, such functional priors implicitly determine priors over BNN weights, and the proposed formulation yields an approximate posterior over the weights from which it is possible to sample through MCMC or any other approximate inference techniques. In this paper, we studied the scenario where functional priors are expressed in the form of Gaussian processes (GPs), but our formulation can handle any functional prior which can be expressed in closed form. We then discussed how to scale our approach to handle large data sets by operating on mini-batches, despite the complications stemming from the use of GP priors.

We tested our proposal on regression and classification tasks and compared it with state-of-the-art approaches to carry out inference and prior optimization for BNNs. Our results demonstrate that the proposed approach is competitive in terms of performance and

quantification of uncertainty, while being easy to implement.

We are currently investigating ways to handle GP priors with priors over hyper-parameters for increased flexibility, and alternative ways to specify functional priors. Furthermore, we are investigating applications of BNNs for image classification tasks for which BNN architectures use convolutional layers.

Acknowledgements

MF gratefully acknowledges support from the AXA Research Fund and the Agence Nationale de la Recherche (grant ANR-18-CE46-0002 and ANR-19-P3IA-0002).

REFERENCES

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.

Chen, T., Fox, E., and Guestrin, C. (2014). Stochastic Gradient Hamiltonian Monte Carlo. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1683–1691, Bejing, China. PMLR.

Cutajar, K., Bonilla, E. V., Michiardi, P., and Filippone, M. (2017). Random feature expansions for deep Gaussian processes. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 884–893. PMLR.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Fortuin, V. (2022). Priors in bayesian deep learning: A review. *International Statistical Review*.

Fortuin, V., Garriga-Alonso, A., Wenzel, F., Ratsch, G., Turner, R. E., van der Wilk, M., and Aitchison, L. (2021). Bayesian neural network priors revisited. In *Third Symposium on Advances in Approximate Bayesian Inference*.

Graves, A. (2011). Practical variational inference for neural networks. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Kallenberg, O. and Kallenberg, O. (1997). *Foundations of modern probability*, volume 2. Springer.

Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. K. (2018). Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ma, C., Li, Y., and Hernandez-Lobato, J. M. (2019). Variational implicit processes. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4222–4233. PMLR.

Milios, D., Camoriano, R., Michiardi, P., Rosasco, L., and Filippone, M. (2018). Dirichlet-based gaussian processes for large-scale calibrated classification. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Neal, R. (1996). Bayesian learning for neural networks. *Lecture Notes in Statistics*.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

Rudner, T. G. J., Chen, Z., and Gal, Y. (2021). Rethinking function-space variational inference in bayesian neural networks. In *Third Symposium on Advances in Approximate Bayesian Inference*.

Shi, J., Sun, S., and Zhu, J. (2018). A spectral approach to gradient estimation for implicit distributions. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4644–4653. PMLR.

Sun, S., Zhang, G., Shi, J., and Grosse, R. (2019). Functional Variational Bayesian Neural Networks. In *International Conference on Learning Representations*.

Tran, B.-H., Rossi, S., Milios, D., and Filippone, M. (2022). All you need is a good functional prior for bayesian deep learning. *Journal of Machine Learning Research*, 23(74):1–56.

Wenk, P., Gotovos, A., Bauer, S., Gorbach, N. S., Krause, A., and Buhmann, J. M. (2019). Fast gaussian process based gradient matching for parameter identification in systems of nonlinear odes. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1351–1360. PMLR.

Wenzel, F., Roth, K., Veeling, B., Swiatkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. (2020). How good is the Bayes posterior in deep neural networks really? In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10248–10259. PMLR.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.