3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–10

Neural Linear Models with Functional Gaussian Process Priors

Joe Watson* JOE.WATSON@TU-DARMSTADT.DE
Jihao Andreas Lin* JIHAOANDREAS.LIN@STUD.TU-DARMSTADT.DE
Pascal Klink PASCAL.KLINK@TU-DARMSTADT.DE
Jan Peters JAN.PETERS@TU-DARMSTADT.DE
Intelligent Autonomous Systems Group, Technical University of Darmstadt

Abstract

Neural linear models (NLM) and Gaussian processes (GP) are both examples of Bayesian linear regression on rich feature spaces. In contrast to the widespread use of nonparametric GPs for probabilistic nonlinear regression, NLMs remain an underused parametric alternative because standard type II maximum likelihood (ML) training leads to overconfidence outside of the data distribution. Therefore, we propose to augment this training procedure through functional variational inference (fVI) proposed by Sun et al. (2019), which is particularly well suited for NLMs due to their closed-form predictive distribution. Additionally, we investigate whether an appropriate functional prior can guide parametric NLMs to attain nonparametric GP performance, despite using fewer parameters. Results show that functional priors can improve performance of NLM over ML training, and that the NLM performs on par with weight space BNNs in this setting.

1. Introduction

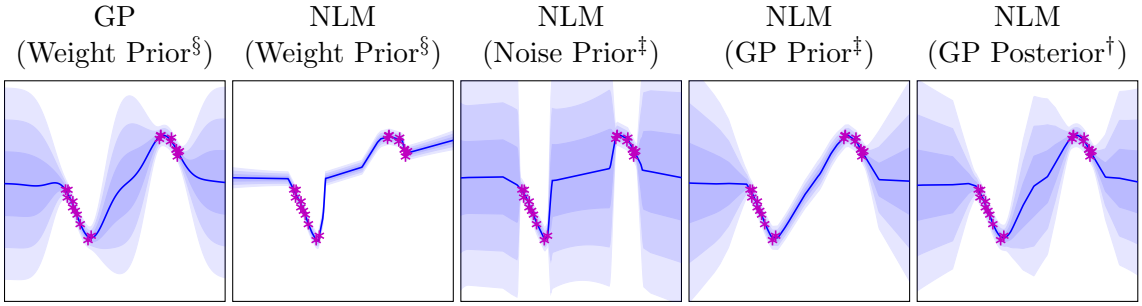

Figure 1: Predictive densities of Bayesian models —— of data *. The models are trained with type II ML (§), functional ELBO (‡) and functional KL (†) objectives, and the NLMs use leaky ReLU activations. Functional priors demonstrate that the NLM is expressive enough to learn diverse feature spaces and approximate a GP, but type II ML and fELBO training can lead to overfitting in the features resulting in reduced OOD uncertainty.

The neural linear model (NLM) (Lázaro-Gredilla and Figueiras-Vidal (2010); Ober and Rasmussen (2019)) is a Bayesian neural network which combines deterministic neural network features with a distibution over the final layer weights. As such, it is a form of

* equal contribution

Bayesian linear regression with a finite feature space, contrasting Gaussian processes (GP) with infinite features spaces induced by kernel functions (Rasmussen and Williams (2005)). These infinite feature spaces make GPs rich probabilistic models for nonlinear regression, which are capable of expressing well-calibrated uncertainty quantification (UQ) outside of the data distribution (OOD). This UQ is crucial for tasks where sample efficiency and safety are critical, such as active learning, Bayesian optimization and model-based reinforcement learning. Bayesian neural networks (BNN) have been widely criticized for their poor UQ (e.g. Foong et al. (2019); Osband et al. (2018)), which impedes their performance in tasks for which Bayesian methods are preferred.

The neural linear model has several attractive qualities compared to alternative BNN approaches, such as Hamiltonian Monte Carlo (Neal (1995)), stochastic variational inference (Blundell et al. (2015)), Monte Carlo dropout (Gal and Ghahramani (2016)) and deep ensembles (Lakshminarayanan et al. (2017)). The predictive distribution is closed-form, rather than an implicit distribution, reducing the computational complexity of prediction. As it is a Bayesian linear model, it also has fewer parameters for equivalent architectures compared to models with weight-space priors. However, as this neural feature space is trained with type II maximum likelihood (ML), the features typically overfit to the data and struggle to express well-calibrated UQ when predicting OOD (Lázaro-Gredilla and Figueiras-Vidal, 2010). Kernel feature spaces demonstrate less susceptibility to this overfitting, as fewer parameters are optimized. Despite this weakness, as GPs struggle to scale as well as BNNs due to their nonparametric nature, we are motivated to enable NLMs that can match GP performance. Therefore, we would wish to assess whether the finite neural feature space is even *capable* of matching or surpassing the performance of Gaussian process kernels.

To investigate this aspect of NLMs, we look at functional variational inference (fVI), which seeks to apply approximate inference over function distributions. With the functional variational BNN (fBNN), Sun et al. (2019) used a GP posterior as a functional prior for BNNs, which was then refined on the data using the functional evidence lower bound objective (fELBO). While this approach is essentially a form of empirical Bayes that requires two Bayesian models to be learned, we seek to use fVI to assess the representational power of NLMs in the context of GPs and other priors. If NLMs are capable of matching or surpassing the performance of nonparametric GPs, then this motivates research into learning objectives for NLMs that encourages more kernel-like feature spaces.

2. Neural Linear Model

Intuitively, NLMs can be viewed as Bayesian linear regression in a projected feature space, where the projection is learned by a neural network. Note that while the NLM can be easily deployed for multivariate regression, the following derivations (and later experiments) in this work focus on univariate targets for simplicity.

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be the observed data, $\mathbf{x}_i \in \mathbb{R}^k$, $y_i \in \mathbb{R}$, such that $\mathbf{X} \in \mathbb{R}^{n \times k}$ and $\mathbf{y} \in \mathbb{R}^n$. Additionally, let $\boldsymbol{\phi}(\cdot; \boldsymbol{\theta}) \colon \mathbb{R}^k \to \mathbb{R}^m$ be a feature space projection with parameters $\boldsymbol{\theta}$, $\boldsymbol{\phi}_i = \boldsymbol{\phi}(\mathbf{x}_i; \boldsymbol{\theta})$ and $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1^\top \dots \boldsymbol{\phi}_n^\top \end{bmatrix} \in \mathbb{R}^{n \times m}$, the matrix of vertically stacked row vectors. It is common to add constant or linear terms to $\boldsymbol{\Phi}$ to implicitly represent bias or identity terms. However, for notational clarity, we ignore these terms and denote the projected feature space as $\mathbb{R}^m$.

A latent function f is modeled using Bayesian linear regression (Box and Tiao (1973); Bishop (2006); Murphy (2012)) with weights $\boldsymbol{\beta}$ and additive, zero-mean, Gaussian noise ϵ with variance σ^2, where

$$y_i = f(\mathbf{x}_i; \boldsymbol{\theta}) = \boldsymbol{\phi}_i^\top \boldsymbol{\beta} + \epsilon_i. \tag{1}$$

Placing a conjugate Gaussian prior $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0^{-1})$ over $\boldsymbol{\beta}$ results in a Gaussian posterior $\mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Lambda}_n^{-1})$ with an explicit Gaussian predictive distribution for query $\mathbf{x}$,

$$y \mid \mathbf{x}, \mathcal{D}, \boldsymbol{\theta} \sim f_p(\mathbf{x}) = \mathcal{N}(\cdot \mid \boldsymbol{\phi}_\mathbf{x}^\top \boldsymbol{\mu}_n, \sigma^2 + \boldsymbol{\phi}_\mathbf{x}^\top \boldsymbol{\Lambda}_n^{-1} \boldsymbol{\phi}_\mathbf{x}), \tag{2}$$

where $\boldsymbol{\mu}_n$ and $\boldsymbol{\Lambda}_n$ are the mean vector and precision matrix of the posterior weight distribution. The exact posteriors are detailed in Section A. The observation noise σ^2, prior weight parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\Lambda}_0$, and $\boldsymbol{\theta}$ can be either set to constants or optimized jointly by maximizing the marginal likelihood. With $\boldsymbol{\mu}_0 = \mathbf{0}$, this model is equivalent to a Gaussian process with kernel $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \boldsymbol{\phi}(\mathbf{x}; \boldsymbol{\theta})^\top \boldsymbol{\Lambda}_0^{-1} \boldsymbol{\phi}(\mathbf{x}'; \boldsymbol{\theta})$ (Rasmussen and Williams (2005)). This connection highlights that the NLM can be viewed as a distribution over functions $p(f)$, in which f is restricted to the form of Equation (1).

3. Functional Variational Inference

Functional variational inference considers optimizing a parameteric posterior $q(f)$ using the functional evidence lower bound objective (fELBO) through use of a functional KL divergence (fKL) and functional prior $p(f)$,

$$\mathcal{L}(\mathcal{D}) := \mathbb{E}_{f \sim q(\cdot)}[\log p(\mathcal{D} \mid f)] - D_{\mathrm{KL}}(q(f) \,||\, p(f)). \tag{3}$$

Here, $\mathcal{L}(\mathcal{D})$ is a lower bound of the log-marginal likelihood $\log p(\mathcal{D})$, such that maximizing $\mathcal{L}(\mathcal{D})$ translates to maximizing $p(\mathcal{D})$. In general, $p(f)$ and $q(f)$ can be arbitrary stochastic processes, i.e. distributions over functions, and the fKL between two stochastic processes (de G. Matthews et al. (2016)) is a generalization of the regular KL divergence between two finite dimensional probability distributions. However, the generalization to function space introduces obstacles in terms of tractability. Roughly speaking, evaluation of the fKL requires integration in an infinite-dimensional vector space, which is intractable as there is no infinite-dimensional Lebesgue measure (Hunt (1992)). Nonetheless, it is possible to approximate this quantity for Gaussian processes by averaging over a finite number of evaluations at s locations $\mathbf{x}_j$ (Sun et al. (2019)),

$$D_{\mathrm{KL}}(q(f) \,||\, p(f)) \approx \tfrac{1}{s} \textstyle\sum_{j=1}^{s} D_{\mathrm{KL}}(f_q(\mathbf{x}_j) \,||\, f_p(\mathbf{x}_j)), \tag{4}$$

such that the quantity is concrete.

A key factor for a successful fKL approximation is the choice of $\{\mathbf{x}_j\}$. Ideally, $\{\mathbf{x}_j\}$ should be representative of the domain of f. In general, it is impossible to entirely cover or reasonably represent the behavior of arbitrary distributions $p(f)$ and $q(f)$ over $\mathbb{R}^k$ with any finite number of samples. Therefore, it is necessary to select a bounded subset of the original domain as 'domain of interest', which can be challenging if the underlying data distribution is unknown.

4. Functional Priors in Practice

While the variational stochastic process $q(f)$ can be modeled using any probabilistic function approximator, such as a NLM or any other Bayesian neural network, a beneficial yet practicable choice for the prior stochastic process $p(f)$ is less apparent. White Gaussian noise (WGN), more specifically a Gaussian process with zero mean and isotropic covariance,

$$p(f) := \mathcal{GP}(\cdot \mid 0, \sigma_p^2), \qquad f_p(\mathbf{x}) := \mathcal{N}(\cdot \mid 0, \sigma_p^2), \tag{5}$$

is a simple candidate which is often used by GPs (Rasmussen and Williams, 2005). However, while GPs typically optimize the variance of this prior using type II ML, here it is fixed. To attain scale invariance across datasets, it is sensible to define this prior in whitened space.

Alternatively, it is possible to use a more informative prior, such as a Gaussian process posterior $\mathcal{GP}(f \mid 0, \mathbf{K})$ whose kernel parameters were calibrated using the available training data (Sun et al. (2019); Shi et al. (2019)). The scalar k, vector $\mathbf{k}$ and matrix $\mathbf{K}$ are computed using the kernel, training data and function input $\mathbf{x}$, so

$$f_p(\mathbf{x}) := \mathcal{N}(\cdot \mid \mathbb{E}(\mathbf{x}), \mathbb{V}(\mathbf{x})), \qquad \mathbb{E}(\mathbf{x}) = \mathbf{k}^\top \mathbf{K}^{-1} \mathbf{y}, \qquad \mathbb{V}(\mathbf{x}) = k - \mathbf{k}^\top \mathbf{K}^{-1} \mathbf{k}. \tag{6}$$

Note that this choice of functional prior $p(f)$ involves the training data $\mathcal{D}$. Therefore, it is not a pure prior in a Bayesian sense, but could be interpreted as a form of empirical Bayes (Morris (1983)), which may result in overfitting.

Lastly, it is also possible to use the fKL solely as the optimization objective, i.e. Equation (4) rather than Equation (3), so that $q(f)$ approximates another stochastic process $p(f)$, e.g. a Gaussian process posterior. If $p(f)$ adequately represents the observed data then $q(f)$ will naturally fit the data by minimizing the fKL while also inheriting other properties, such as extrapolation behavior and uncertainty quantification. In particular, this functional inference can transfer stochastic processes which are modeled by arbitrary probabilistic function approximators, e.g. GPs and NLMs, even if both models have considerably different architectures. This could enable model compression and distillation, which may be practical for nonparametric models like Gaussian processes.

The computational complexity of the training procedure for NLMs with the fELBO is of important practical concern and hence we discuss it here. The prediction complexity is, regardless of the chosen training procedure, in $\mathcal{O}(m^2)$ per prediction, assuming the feature-space dimension m to be the largest of all hidden-layer dimensions and neglecting the computation of the non-linearities of the network. The total complexity during training of evaluating the fELBO (Equation (3)) is the sum of the complexities of evaluating the marginal likelihood and the fKL approximation (Equation (4)). Under the previous assumptions, the complexity of the marginal likelihood objective for the NLM is in $\mathcal{O}(nm^2+m^3)$. The complexity of computing the fKL approximation depends on the number of samples s and the types of involved stochastic processes. Assuming that f_p and f_q are both Gaussian and already available, computation of the fKL term with s samples $\{\mathbf{x}_j\}$ requires $\mathcal{O}(sd^3)$ for d output dimensions, which simplifies to $\mathcal{O}(s)$ for $d = 1$. Computing f_q consists of evaluating the NLM's predictive distribution, which is in $\mathcal{O}(sm^2)$ for s samples. The complexity of computing f_p depends on the chosen prior. The WGN prior is available in $\mathcal{O}(1)$ time, whereas the GP prior requires GP training once ($\mathcal{O}(n^3)$) and GP prediction ($\mathcal{O}(sn^2)$) per objective evaluation.

	MODEL		BOSTON	CONCRETE	ENERGY	WINE	YACHT
GP	§	RBF	-2.516 ± 0.114	-3.512 ± 0.011	$\mathbf{-0.659 \pm 0.050}$	$\mathbf{-0.963 \pm 0.021}$	$\mathbf{-0.150 \pm 0.037}$
NLM	§	LRELU	-2.563 ± 0.048	-3.105 ± 0.026	-0.905 ± 0.058	-1.000 ± 0.013	-1.325 ± 0.087
		TANH	-2.634 ± 0.043	-3.107 ± 0.019	-1.067 ± 0.077	-0.991 ± 0.011	-0.954 ± 0.082
	WGN ‡	LRELU	-2.569 ± 0.069	-3.167 ± 0.014	-1.126 ± 0.017	$\mathbf{-0.981 \pm 0.018}$	-1.592 ± 0.075
		TANH	-2.570 ± 0.066	-3.093 ± 0.015	-1.381 ± 0.024	$\mathbf{-0.989 \pm 0.017}$	-1.100 ± 0.109
	GP ‡	LRELU	-2.538 ± 0.087	$\mathbf{-3.010 \pm 0.024}$	$\mathbf{-0.627 \pm 0.063}$	$\mathbf{-0.976 \pm 0.030}$	-0.785 ± 0.099
		TANH	-2.594 ± 0.121	$\mathbf{-3.015 \pm 0.037}$	-0.748 ± 0.046	$\mathbf{-0.994 \pm 0.022}$	-0.697 ± 0.077
	GP †	LRELU	-2.604 ± 0.140	-3.449 ± 0.005	$\mathbf{-0.660 \pm 0.036}$	$\mathbf{-0.959 \pm 0.021}$	-0.886 ± 0.083
		TANH	-2.533 ± 0.126	-3.452 ± 0.007	$\mathbf{-0.633 \pm 0.041}$	$\mathbf{-0.991 \pm 0.019}$	-0.723 ± 0.072
BBB[1]	∗	RELU	-2.602 ± 0.031	-3.149 ± 0.018	-1.500 ± 0.006	$\mathbf{-0.977 \pm 0.017}$	-2.408 ± 0.007
NNG[2]	∗	RELU	-2.446 ± 0.029	$\mathbf{-3.039 \pm 0.025}$	-1.421 ± 0.005	$\mathbf{-0.969 \pm 0.014}$	-2.316 ± 0.006
fBNN[3]	GP ‡	RELU	$\mathbf{-2.301 \pm 0.038}$	-3.096 ± 0.016	$\mathbf{-0.684 \pm 0.020}$	-1.040 ± 0.013	-1.033 ± 0.033

(*a*) Log-Likelihood

	MODEL		BOSTON	CONCRETE	ENERGY	WINE	YACHT
GP	§	RBF	2.977 ± 0.249	5.540 ± 0.162	0.472 ± 0.021	$\mathbf{0.628 \pm 0.019}$	$\mathbf{0.363 \pm 0.028}$
NLM	§	LRELU	3.055 ± 0.178	$\mathbf{5.007 \pm 0.124}$	0.459 ± 0.015	$\mathbf{0.649 \pm 0.010}$	0.922 ± 0.094
		TANH	3.195 ± 0.154	$\mathbf{5.127 \pm 0.133}$	0.523 ± 0.017	$\mathbf{0.639 \pm 0.010}$	0.576 ± 0.067
	WGN ‡	LRELU	3.010 ± 0.228	$\mathbf{5.043 \pm 0.156}$	0.458 ± 0.021	$\mathbf{0.641 \pm 0.013}$	1.189 ± 0.137
		TANH	2.984 ± 0.194	$\mathbf{4.897 \pm 0.156}$	0.544 ± 0.026	$\mathbf{0.646 \pm 0.013}$	0.633 ± 0.120
	GP ‡	LRELU	2.982 ± 0.226	$\mathbf{4.833 \pm 0.140}$	$\mathbf{0.437 \pm 0.020}$	$\mathbf{0.638 \pm 0.017}$	0.502 ± 0.053
		TANH	3.026 ± 0.223	$\mathbf{4.828 \pm 0.211}$	0.503 ± 0.019	$\mathbf{0.652 \pm 0.015}$	0.480 ± 0.047
	GP †	LRELU	2.962 ± 0.246	5.536 ± 0.130	0.463 ± 0.018	$\mathbf{0.627 \pm 0.015}$	0.471 ± 0.049
		TANH	2.830 ± 0.255	5.654 ± 0.144	0.453 ± 0.018	$\mathbf{0.648 \pm 0.014}$	0.453 ± 0.047
BBB[1]	∗	RELU	3.171 ± 0.149	5.678 ± 0.087	0.565 ± 0.018	$\mathbf{0.643 \pm 0.012}$	1.174 ± 0.086
NNG[2]	∗	RELU	2.742 ± 0.125	$\mathbf{5.019 \pm 0.127}$	0.485 ± 0.023	$\mathbf{0.637 \pm 0.011}$	0.979 ± 0.077
fBNN[3]	GP ‡	RELU	$\mathbf{2.378 \pm 0.104}$	$\mathbf{4.935 \pm 0.180}$	$\mathbf{0.412 \pm 0.017}$	0.673 ± 0.014	0.607 ± 0.068

(*b*) RMSE

Table 1: Regression results on the UCI dataset. Models are trained with a type II maximum likelihood (§), ELBO (∗), functional ELBO (‡) and functional KL (†) objective. Results obtained from 1, Blundell et al. (2015); 2, Zhang et al. (2018); 3, Sun et al. (2019).

5. Experimental Results

This section discusses our nonlinear regression experiment, which aims to demonstrate the viability of combining NLMs with functional inference, leveraging the NLM's explicit predictive distribution. To this end, we replicated the experiments presented in Sun et al. (2019) and evaluated the NLM on several UCI regression datasets. The goal is to assess how the NLMs with functional priors compare to other Bayesian neural networks with weight space priors. In this setting, four different optimization objectives, namely type II maximum likelihood with the conventional weight space prior (§), weight space ELBO (∗), functional ELBO (‡) and functional KL (†), were used. For the functional inference, we used white Gaussian noise (WGN) and a Gaussian process (GP) as functional priors.

All models were implemented using PyTorch (Paszke et al. (2019)). For the GPs we additionally used GPyTorch (Gardner et al. (2018)). The NLM was implemented with $\boldsymbol{\mu}_0 = \mathbf{0}$ and a diagonal matrix with m distinct parameters to represent $\boldsymbol{\Lambda}_0$. The observation noise parameter σ^2, neural network weights $\boldsymbol{\theta}$ and $\boldsymbol{\Lambda}_0$ were learned via backpropagation.

To encode positive value constraints, σ^2 and $\mathbf{\Lambda}_0$ were learned in log-space. Following Sun et al. (2019), NLMs used a single hidden layer with 50 units. For optimization parameters, please refer to Appendix B.

Table 1 displays the results of our UCI regression benchmark experiments, comparing NLMs to GPs and previous work. In general, the NLM achieves competitive performance. Although using the WGN prior results in underfitting for data with low observation noise (energy, yacht), it can match or surpass the predictive performance of a GP on data with high observation noise (boston, concrete, wine). Applying fELBO inference with the GP prior, the NLM mostly outperforms the other Bayesian neural networks. The fKL objective with the GP posterior has performance that does indeed match the GP across most datasets. The NLM with fELBO training also generally outperforms the fBNN, apart from the boston dataset. Given that the fBNN has approximately twice as many parameters, a more elaborate training procedure, and an implicit predictive distribution, this indicates that the NLM is a better function class for functional prior research.

6. Related Work

The study of functional priors for Bayesian inference is a burgeoning field. Our work builds off Sun et al. (2019), who train a weight-space BNN with a GP posterior using a KL gradient estimator and mean-field stochastic variational inference. Shi et al. (2019) in turn build off this, taking a mirror descent interpretation to allow for incremental training with minibatches and the Gaussian process prior. As Gaussian processes are exact distributions over functions, sparse GPs may be viewed as approximate inference over functions (de G. Matthews et al. (2016)), minimizing the fKL from its exact posterior via inducing points. The noise contrastive prior (NCP) of Hafner et al. (2019) is a similar idea where the training data is perturbed by random noise to serve as a 'data prior' for a BNN, in order to increase uncertainty estimation OOD. While effective empirically, the data prior is again akin to empirical Bayes as the prior is defined by the data. The practice of combining kernels in GPs has been translated to BNN architectures and activation functions, producing periodic and mixing phenomena in the network's feature space for more expressive models (Pearce et al. (2019)). Variational implicit priors (Ma et al. (2019)) use variational inference to deal with functional priors which you can only sample from, e.g. simulators, which provides the flexibility of a broad range of complex processes to be adopted as priors.

7. Outlook

The goal of this work is to motivate further research into the NLM by demonstrating its potential in combination with functional variational inference. Experiments have shown that this model is just as capable of representing a GP posterior as networks with weight priors, while being simpler and having closed-form predictive distributions. Our results suggest that an appropriate functional prior could enable GP-like performance and uncertainty quantification from NLMs, without requiring a GP posterior for training.

Acknowledgments

Pascal Klink is funded by the DFG project PA3179/1-1 (ROBOLEAP).

References

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, Berlin, Heidelberg, 2006.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, 2015.

G. E. P. Box and G. C. Tiao. *Bayesian Inference in Statistical Analysis*. John Wiley & Sons, New York, 1973.

Alexander G. de G. Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. On sparse variational methods and the kullback-leibler divergence between stochastic processes. In *International Conference on Artificial Intelligence and Statistics*, 2016.

Andrew Foong, Yingzhen Li, José Hernández-Lobato, and Richard Turner. 'in-between' uncertainty in bayesian neural networks. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2016.

Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

Danijar Hafner, Dustin Tran, Alex Irpan, Timothy Lillicrap, and James Davidson. Noise contrastive priors for functional uncertainty. In *Uncertainty in Artificial Intelligence*, 2019.

B. Hunt. Prevalence: a translation-invariant "almost every" on infinite-dimensional spaces. *Bulletin of the American Mathematical Society*, 1992.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

Miguel Lázaro-Gredilla and Aníbal R. Figueiras-Vidal. Marginalized neural network mixtures for large-scale regression. *Transactions on Neural Networks*, 21(8), 2010.

Chao Ma, Yingzhen Li, and Jose Miguel Hernandez-Lobato. Variational implicit processes. In *International Conference on Machine Learning*, 2019.

Carl N Morris. Parametric empirical bayes inference: theory and applications. *Journal of the American statistical Association*, 78(381), 1983.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

Radford M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, CAN, 1995.

Sebastian W. Ober and Carl Edward Rasmussen. Benchmarking the neural linear model for regression. In *Symposium on Advances in Approximate Bayesian Inference*, 2019.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

Tim Pearce, Russell Tsuchida, Mohamed Zaki, Alexandra Brintrup, and Andy Neely. Expressive priors in bayesian neural networks: Kernel combinations and periodic functions. In *Uncertainty in Artificial Intelligence*, 2019.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

Jiaxin Shi, Mohammad Emtiyaz Khan, and Jun Zhu. Scalable training of inference networks for Gaussian-process models. In *International Conference on Machine Learning*, 2019.

Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational bayesian neural networks. In *International Conference on Learning Representations*, 2019.

Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, 2018.

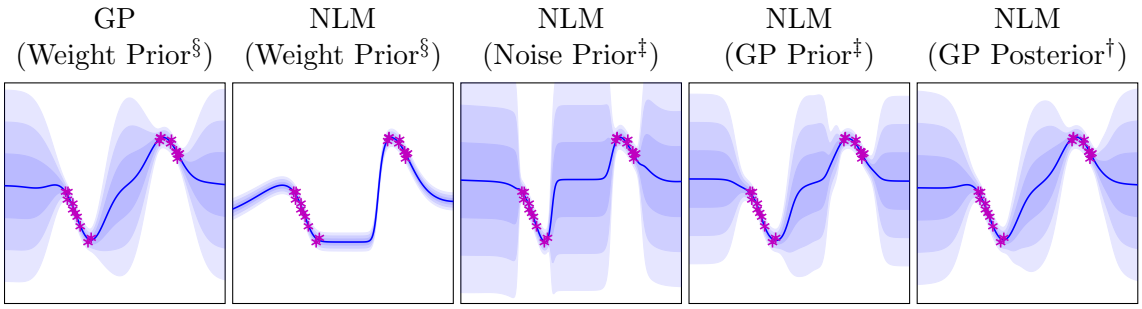

Figure 2: Reproduction of Figure 1 with tanh activation for the NLM.

Appendix A. Bayesian Last Layer Equations

In this section, we collectively state the equations for Bayesian linear regression (Box and Tiao (1973); Bishop (2006); Murphy (2012)) with weights $\boldsymbol{\beta}$ and additive noise ϵ, where

$$y_i = f(\mathbf{x}_i; \boldsymbol{\theta}) = \boldsymbol{\phi}_i^\top \boldsymbol{\beta} + \epsilon_i, \tag{7}$$

$$\epsilon_i \sim \mathcal{N}(\cdot \mid 0, \sigma^2), \tag{8}$$

$$\mathbf{y} \sim \mathcal{N}(\cdot \mid \boldsymbol{\Phi}\boldsymbol{\beta}, \sigma^2 \mathbf{I}). \tag{9}$$

Assuming known aleatoric uncertainty σ^2 (Bishop (2006)), a conjugate Gaussian prior over $\boldsymbol{\beta}$ results in a Gaussian posterior,

$$\boldsymbol{\beta} \sim \mathcal{N}(\cdot \mid \boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0^{-1}), \qquad \boldsymbol{\mu}_n = \boldsymbol{\Lambda}_n^{-1}(\boldsymbol{\Lambda}_0 \boldsymbol{\mu}_0 + \sigma^{-2} \boldsymbol{\Phi}^\top \mathbf{y}), \tag{10}$$

$$\boldsymbol{\beta} \mid \mathcal{D}, \boldsymbol{\theta} \sim \mathcal{N}(\cdot \mid \boldsymbol{\mu}_n, \boldsymbol{\Lambda}_n^{-1}), \qquad \boldsymbol{\Lambda}_n = \sigma^{-2} \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \boldsymbol{\Lambda}_0, \tag{11}$$

with an explicit Gaussian predictive distribution for output y and query $\mathbf{x}$,

$$y \mid \mathbf{x}, \mathcal{D}, \boldsymbol{\theta} \sim \mathcal{N}(\cdot \mid \boldsymbol{\phi}_\mathbf{x}^\top \boldsymbol{\mu}_n, \sigma^2 + \boldsymbol{\phi}_\mathbf{x}^\top \boldsymbol{\Lambda}_n^{-1} \boldsymbol{\phi}_\mathbf{x}). \tag{12}$$

The log-marginal likelihood can be written as

$$\log p(\mathcal{D} \mid \boldsymbol{\theta}) = \frac{1}{2\sigma^2}(\boldsymbol{\mu}_n^\top \boldsymbol{\Lambda}_n \boldsymbol{\mu}_n - \boldsymbol{\mu}_0^\top \boldsymbol{\Lambda}_0 \boldsymbol{\mu}_0 - \mathbf{y}^\top \mathbf{y}) + \frac{1}{2} \log \frac{|\boldsymbol{\Lambda}_0|}{|\boldsymbol{\Lambda}_n|} - \frac{n}{2} \log 2\pi\sigma^2. \tag{13}$$

Appendix B. Optimization Parameters

Gradient-based optimization was performed using Adam (Kingma and Ba (2014)). The GPs were trained until convergence for up to 3000 epochs at a learning rate of 1e-2. The NLMs were trained at a learning rate of 1e-3, and the number of training epochs was determined by tracking the validation log-likelihood of 10% of the training data for up to 10000 epochs for ML training and up to 15000 epochs for all fVI variants. The fKL was evaluated using 100 samples per epoch from a uniform distribution consistent with Sun et al. (2019).

Appendix C. Experimental Results

MODEL			BOSTON	CONCRETE	ENERGY	WINE	YACHT
GP	§	RBF	$\mathbf{-1.980 \pm 0.013}$	-3.439 ± 0.004	-0.310 ± 0.007	$\mathbf{-0.718 \pm 0.003}$	$\mathbf{0.415 \pm 0.017}$
NLM	§	LRELU	-2.180 ± 0.026	-2.860 ± 0.042	-0.728 ± 0.085	-0.864 ± 0.014	-1.008 ± 0.086
		TANH	-2.290 ± 0.037	-2.842 ± 0.024	-0.973 ± 0.088	-0.906 ± 0.008	-0.810 ± 0.053
	WGN ‡	LRELU	-2.145 ± 0.037	-3.039 ± 0.012	-1.089 ± 0.020	-0.871 ± 0.004	-1.284 ± 0.072
		TANH	-2.169 ± 0.038	-2.917 ± 0.009	-1.362 ± 0.023	-0.925 ± 0.007	-0.894 ± 0.059
	GP ‡	LRELU	-2.072 ± 0.024	$\mathbf{-2.759 \pm 0.017}$	$\mathbf{-0.281 \pm 0.016}$	-0.759 ± 0.019	-0.475 ± 0.073
		TANH	-2.074 ± 0.030	$\mathbf{-2.761 \pm 0.025}$	-0.582 ± 0.010	-0.822 ± 0.017	-0.531 ± 0.028
	GP †	LRELU	-2.017 ± 0.019	-3.389 ± 0.003	-0.442 ± 0.014	-0.825 ± 0.004	-0.706 ± 0.102
		TANH	-2.057 ± 0.027	-3.385 ± 0.004	-0.416 ± 0.007	-0.816 ± 0.009	-0.524 ± 0.044

(*a*) Log-Likelihood

MODEL			BOSTON	CONCRETE	ENERGY	WINE	YACHT
GP	§	RBF	1.505 ± 0.021	3.464 ± 0.024	$\mathbf{0.309 \pm 0.002}$	$\mathbf{0.378 \pm 0.003}$	$\mathbf{0.119 \pm 0.002}$
NLM	§	LRELU	$\mathbf{1.239 \pm 0.032}$	3.087 ± 0.062	0.342 ± 0.007	0.527 ± 0.012	0.486 ± 0.029
		TANH	1.353 ± 0.034	3.053 ± 0.052	0.451 ± 0.008	0.560 ± 0.009	0.404 ± 0.023
	WGN ‡	LRELU	$\mathbf{1.182 \pm 0.053}$	3.315 ± 0.063	0.353 ± 0.006	0.555 ± 0.003	0.548 ± 0.041
		TANH	1.301 ± 0.055	2.877 ± 0.045	0.470 ± 0.012	0.596 ± 0.004	0.345 ± 0.019
	GP ‡	LRELU	1.303 ± 0.019	$\mathbf{2.744 \pm 0.070}$	$\mathbf{0.306 \pm 0.006}$	0.498 ± 0.010	0.188 ± 0.007
		TANH	1.323 ± 0.027	$\mathbf{2.775 \pm 0.073}$	0.428 ± 0.004	0.533 ± 0.009	0.370 ± 0.019
	GP †	LRELU	1.582 ± 0.023	4.019 ± 0.024	0.336 ± 0.003	0.521 ± 0.003	0.223 ± 0.013
		TANH	1.583 ± 0.022	3.984 ± 0.027	0.346 ± 0.002	0.502 ± 0.009	0.287 ± 0.025

(*b*) RMSE

Table 2: Training metrics on the UCI dataset. Models are trained with a type II maximum likelihood (§), ELBO (∗), functional ELBO (‡) and functional KL (†) objective. Training results of Blundell et al. (2015); Zhang et al. (2018); Sun et al. (2019) are omitted as they were not reported.