2nd Symposium on Advances in Approximate Bayesian Inference, 2019 1–10

Gaussian Process Meta-Representations For Hierarchical Neural Network Weight Priors

Theofanis Karaletsos* THEOFANIS@UBER.COM
Thang D. Bui* THANG.BUI@UBER.COM
Uber AI, San Francisco, CA, USA

Abstract

Bayesian inference offers a theoretically grounded and general way to train neural networks and can potentially give calibrated uncertainty. However, it is challenging to specify a meaningful and tractable prior over the network parameters, and deal with the weight correlations in the posterior. To this end, this paper introduces two innovations: (i) a Gaussian process-based hierarchical model for the network parameters based on recently introduced unit embeddings that can flexibly encode weight structures, and (ii) input-dependent contextual variables for the weight prior that can provide convenient ways to regularize the function space being modeled by the network through the use of kernels. We show these models provide desirable test-time uncertainty estimates, demonstrate cases of modeling inductive biases for neural networks with kernels and demonstrate competitive predictive performance on an active learning benchmark.

1. Introduction

The question of which priors one should use for Bayesian neural networks is largely unanswered, as two considerations need to be balanced: First, we want to keep inference in the high dimensional weight posterior tractable; Second, we desire to express our beliefs about the properties of the modeled functions compactly by modeling the collection of weights. Especially the latter is typically hard, as functional regularization for weight-based models is non-trivial. In order to cope with richer posterior inference than mean-field typically achieves, a variety of structured posterior models have been proposed recently, for instance utilizing radial posteriors (Oh et al., 2019), or rich weight posteriors based on Gaussian processes (Louizos and Welling, 2016). When it comes to modeling priors on weights with correlations, recent work has attempted to capture feature-level correlations using for instance a horseshoe prior (Ghosh et al., 2018). One interesting direction of inquiry has focused on utilizing hyper-networks in order to model distributions over weights for an entire network (Ha et al., 2016; Pradier et al., 2018), or alternatively to utilize unit-level level variables combined with compact hyper-networks to regress to single weights and capture weight correlations through the auxiliary variables (Karaletsos et al., 2018). We propose to tackle some of the challenges in modeling weight priors by extending the latter work and combining it with ideas from the Gaussian process literature to replace the hyper-network with a Gaussian process prior over weights. We explore the use of compositional kernels to add input-dependence to the prior for our model and obtain rich models with beneficial

properties in tasks such as active learning, and generalization, while maintaining tractable inference properties.

2. Meta-representing weights and networks

In (Karaletsos et al., 2018) each unit (visible or hidden) of the l-th layer of the network has a corresponding latent hierarchical variable $\mathbf{z}_{l,i}$, of dimensions D_z, where i denotes the index of the unit in a layer. These latent variables are used to construct the weights in the network such that a weight in the l-th weight layer, $w_{l,i,j}$ is linked to the latent variables z's of the i-th input unit and the j-th output unit of the weight layer. We can summarize this relationship by introducing a set of weight encodings, $\mathbf{C}_w(\mathbf{z})$, one for each individual weight, $\mathbf{c}_{w_{l,i,j}} = \left[\mathbf{z}_{l+1,i}, \mathbf{z}_{l,j}\right]$. The probabilistic description of the relationship between the weight codes and the weights $\mathbf{w}$ is: $p(\mathbf{w}|\mathbf{z}) = p(\mathbf{w}|\mathbf{C}_w(\mathbf{z})) = \prod_{l=1}^{L-1}\prod_{i=1}^{H_l}\prod_{j=1}^{H_{l+1}} p(w_{l,i,j}|\mathbf{z}_{l+1,i}, \mathbf{z}_{l,j})$, where l denotes a visible or hidden layer and H_l is the number of units in that layer, and $\mathbf{w}$ denotes all the weights in this network. In (Karaletsos et al., 2018), a small parametric neural network regression model maps the latent variables to the weights, $p(w_{l,i,j}|\mathbf{z}_{l+1,i}, \mathbf{z}_{l,j}, \theta) = \mathcal{N}(w_{l,i,j}; \mu_{l,i,j}, \sigma^2_{l,i,j})$, where $(\mu_{l,i,j}, \log\sigma_{l,i,j}) = \mathrm{NN}_\theta([\mathbf{z}_{l+1,i}, \mathbf{z}_{l,j}])$. We will call this network a *meta mapping*. We assume $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$. We can thus write down the joint density of the resulting hierarchical model as follows,

$$p(\mathbf{y}, \mathbf{w}, \mathbf{z}|\mathbf{x}, \theta) = \left[\prod_{l=1}^{L} p(\mathbf{z}_l)\right] \left[p(\mathbf{w}|\mathbf{C}_w(\mathbf{z}), \theta)\right] \left[\prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{w}, \mathbf{x}_n)\right]. \tag{1}$$

Variational inference was employed in prior work to infer $\mathbf{z}$ (and $\mathbf{w}$ implicitly), and to obtain a point estimate of θ, as a by-product of optimising the variational lower bound.

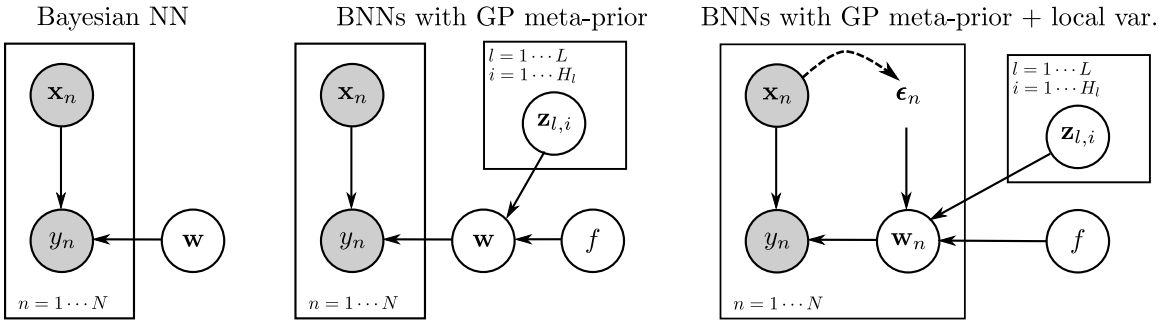

Figure 1: Graphical depiction of various models: vanilla BNNs, BNNs with hierarchical GP-MetaPriors, and BNNs with hierarchical GP-MetaPriors and auxiliary variables.

3. Meta-representing weights with Gaussian processes

Notice that in Sec.2, the meta mapping from the hierarchical latent variables to the weights is a parametric non-linear function, specified by a neural network. We replace the parametric

neural network by a probabilistic functional mapping and place a nonparametric Gaussian process (GP) prior over this function. That is,

$$p(w_{l,i,j}|f, \mathbf{c}_{w_{l,i,j}}) = \mathcal{N}(w_{l,i,j}; f([\mathbf{z}_{l+1,i}, \mathbf{z}_{l,j}]), \sigma_w^2); p(f|\gamma) = \mathcal{GP}(f; \mathbf{0}, k_{c_w}(\cdot, \cdot|\gamma)),$$

where we have assumed a zero-mean GP, $k_\gamma(\cdot, \cdot)$ is a covariance function and γ is a small set of hyper-parameters. The effect is that the latent function introduces correlations for the individual weight predictions,

$$P(\mathbf{w}|\mathbf{z}) = P(\mathbf{w}|\mathbf{C}_w(\mathbf{z})) = \int p(f)\Big[\prod_{l=1}^{L-1}\prod_{i=1}^{H_{l+1}}\prod_{j=1}^{H_l} p(w_{l,i,j}|f, \mathbf{z}_{l+1,i}, \mathbf{z}_{l,j})\Big]df. \tag{2}$$

Notably, while the number of latent variables and weights can be large, the input dimension to the GP mapping is only $2D_z$, where D_z is the dimensionality of each latent variable $\mathbf{z}$. The GP mapping effectively performs one-dimensional regression from latent variables to individual weights while capturing their correlations. We will refer to this mapping as a **GP-MetaPrior** (*metaGP*). We define the following factorized kernel at the example of two weights in the network,

$$k_{c_w}(c_{w^1}, c_{w^2}) = k([\mathbf{z}_{l^1+1,i^1}, \mathbf{z}_{l^1,j^1}], [\mathbf{z}_{l^2+1,i^2}, \mathbf{z}_{l^2,j^2}]) = k_{out}(\mathbf{z}_{l^1+1,i^1}, \mathbf{z}_{l^2+1,i^2}) \cdot k_{in}(\mathbf{z}_{l^1,j^1}, \mathbf{z}_{l^2,j^2}).$$

In this section and what follows, we will use the popular exponentiated quadratic (EQ) kernel with ARD lengthscales, $k(\mathbf{x}_1, \mathbf{x}_2) = \sigma_k^2 \exp\left(\sum_{d=1}^{2D_z} \frac{-(x_{1,d}-x_{2,d})^2}{2l_d^2}\right)$, where $\{l_d\}_{d=1}^{2D_z}$ are the lengthscales and σ_k^2 is the kernel variance. We cover inference and learning in App. A.

3.1. Contextual variables for modulating function priors

We first note that whilst the hierarchical latent variables and meta mappings introduce non-trivial coupling between the weights a priori, the weights and latent variables are inherently global. That is, a function drawn from the model, represented by a set of weights, does not take into account the inputs at which the function will be evaluated. To this end, we introduce the input variable into the weight codes $c_{w_{l,i,j}} = \big[\mathbf{z}_{l+1,i}, \mathbf{z}_{l,j}, \mathbf{x}_n\big]$. In turn, this yields input-conditional weight models $p(w_{n,l,i,j}|f, \mathbf{z}_{l+1,i}, \mathbf{z}_{l,j}, \mathbf{x}_n)$. We again turn to compositional kernels and introduce a new **input kernel** K_x which we use as follows,

$$k_{c_w}(c_{w^1}, c_{w^2}) = k_{out}(\mathbf{z}_{l^1+1,i^1}, \mathbf{z}_{l^2+1,i^2}) \cdot k_{in}(\mathbf{z}_{l^1,j^1}, \mathbf{z}_{l^2,j^2}) \cdot k_x(x_1, x_2).$$

As a result of having private contextual inputs to the meta mapping, the weight priors are now also local to each data point. We can utilize multiple useful kernels from the GP literature that allow modelers to describe relationships between data, but were previously inaccessible to neural network modelers. We consider this a novel form of functional regularization, as the entire network can be given structure that will constrain its function space. To scale this to large inputs, we learn transformations of inputs for the conditional weight model $\boldsymbol{\epsilon}_n = g(\mathbf{V}\mathbf{x}_n)$, for a learned mapping $\mathbf{V}$ and a nonlinearity g:

$$p(w_{n,l,i,j}|f, \mathbf{z}_{l+1,i}, \mathbf{z}_{l,j}, \mathbf{x}_n, \mathbf{V}) = \mathcal{N}(w_{n,l,i,j}; f([\mathbf{z}_{l+1,i}, \mathbf{z}_{l,j}, \boldsymbol{\epsilon}_n]), \sigma_w^2).$$

3.2. Bayesian Neural Networks with GP-MetaPriors

We write down the joint density of all variables in the model when using our weight prior in a neural network:

$$\begin{aligned} p(\mathbf{y}, \mathbf{w}, \mathbf{z}, f|\mathbf{x}) &= p(\mathbf{z})p(f)p(\mathbf{w}|f, \mathbf{z})p(\mathbf{y}|\mathbf{w}, \mathbf{x}) \\ &= p(\mathbf{z})p(f) \prod_{n=1}^{N} \left[p(\mathbf{w}_n|f, \mathbf{C}_w(\mathbf{z}))p(\mathbf{y}_n|\mathbf{w}, \mathbf{x}_n)\right]. \end{aligned}$$

We discuss inference and learning in the Appendix Sec. A.

4. Experiments

We study our suggested priors empirically in two distinct settings in the following: first, we study the effect of kernel choice in the local model for a regression problem where we may have available intuitions as inductive biases. Second, we explore how the input-dependence behaves in out of distribution generalization tasks.

4.1. Inductive Biases For Neural Networks With Input-Dependent Kernels

We explore the utility of the contextual variable towards modeling inductive biases for neural networks and evaluate on predictive performance on a regression example. In particular, we generate 100 training points from a synthetic sinusoidal function and create two test sets that contains in-sample inputs and out-of-sample inputs, respectively. We test an array of models and inference methods, including BNN with MFVI, metaGP and metaGP with contextual variables. We can choose the covariance function to be used for the auxiliary variables to encode our belief about how the weights should be modulated by the input. We pick EQ and periodic kernels (MacKay, 1998) in this example. Fig. 2 summarizes the results and illustrate the qualitative difference between models. Note that the periodic kernel allows the model to discover and encode periodicity, allowing for more long-range confident predictions compared to that of the EQ kernel.

4.2. Input Dependent Neural Networks For Uncertainty Quantification

We test the ability of this model class to produce calibrated predictive uncertainty to out-of-distribution samples. We first train a neural network classifier with one hidden layer of 100 rectified linear units on the MNIST dataset, and apply the metaGP prior only to the last layer of the network. After training, we compute the entropy of the predictions on various test sets, including notMNIST, fashionMNIST, Kuzushiji-MNIST, and uniform and Gaussian noise inputs. Following (Lakshminarayanan et al., 2017; Louizos and Welling, 2017), the CDFs of the predictive entropies for various methods are shown in Fig. 3. In most out-of-distribution sets considered, metaGP and metaGP with local auxiliary variables demonstrate competitive performance to Gaussian MFVI. Notably, MAP estimation tends to give wildly poor uncertainty estimates on out-of-distribution samples.

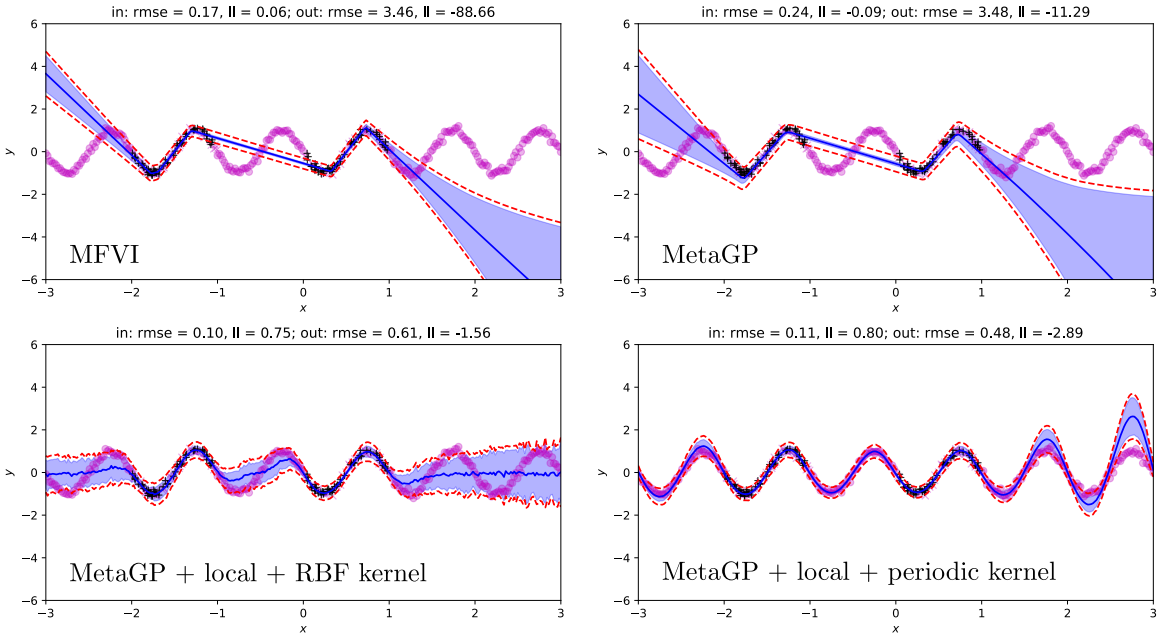

Figure 2: Illustration of the effect of local variables and different kernels for these variables.

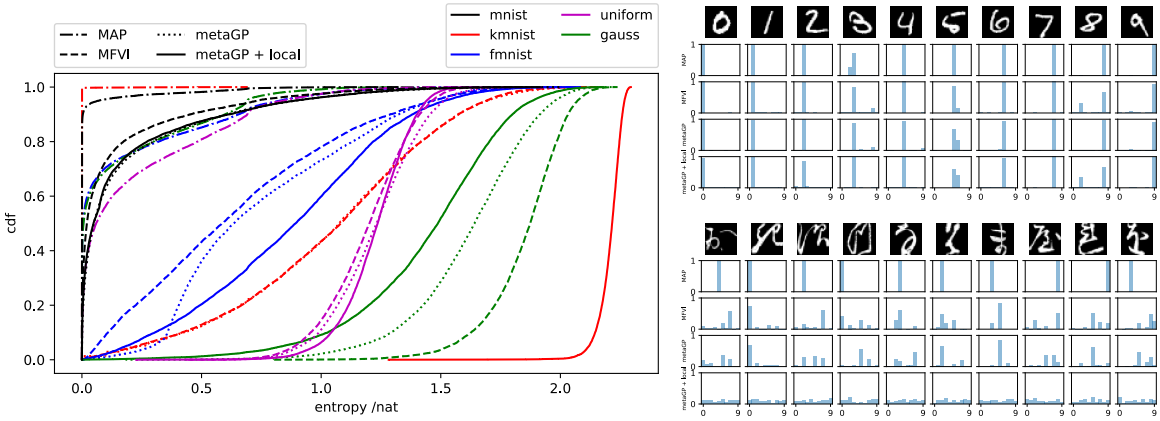

Figure 3: The CDFs of predictive entropies on in-distribution and out-of-distribution test sets for various methods [Left] and the predictive class probability for representative samples from in-distribution/out-of-distribution test sets [Right].

5. Summary

We illustrated the utility of a GP-based hierarchical prior over neural network weights and a variational inference scheme that captures weight correlations and allows input-dependent contextual variables. We plan to evaluate the performance of the model on more challenging decision making tasks and to extend the inference scheme to handle continual learning.

References

Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems*, pages 1533–1541, 2016.

Thang D Bui, Josiah Yan, and Richard E Turner. A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation. *The Journal of Machine Learning Research*, 18(1):3649–3720, 2017.

Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Structured variational learning of Bayesian neural networks with horseshoe priors. In *International Conference on Machine Learning*, pages 1739–1748, 2018.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning*, 2015.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Theofanis Karaletsos, Peter Dayan, and Zoubin Ghahramani. Probabilistic meta-representations of neural networks. *arXiv preprint arXiv:1810.00555*, 2018.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix Gaussian posteriors. In *International Conference on Machine Learning*, pages 1708–1716, 2016.

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational Bayesian neural networks. In *International Conference on Machine Learning*, pages 2218–2227, 2017.

David J.C. MacKay. Introduction to Gaussian processes. *NATO ASI series. Series F: computer and system sciences*, pages 133–165, 1998.

Alexander G. D. G. Matthews, James Hensman, Richard E 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.

Changyong Oh, Kamil Adamczewski, and Mijung Park. Radial and directional posteriors for bayesian neural networks. *arXiv preprint arXiv:1902.02603*, 2019.

Melanie F Pradier, Weiwei Pan, Jiayu Yao, Soumya Ghosh, and Finale Doshi-Velez. Latent projection bnns: Avoiding weight-space pathologies by learning latent representations of neural network weights. *arXiv preprint arXiv:1811.07006*, 2018.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6 (Dec):1939–1959, 2005.

Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009.

Appendix A. Appendix: Inference and learning using stochastic structured variational inference

Performing inference is challenging due to the non-linearity of the neural network and the need to infer an entire latent function f. To address these problems, we derive a structured variational inference scheme that makes use of innovations from inducing point GP approximation literature (Titsias, 2009; Hensman et al., 2013; Quiñonero-Candela and Rasmussen, 2005; Matthews et al., 2016; Bui et al., 2017) and previous work on inferring meta-representations (Karaletsos et al., 2018). As a reminder, we write down the joint density of all variables in the model:

$$\begin{aligned} p(\mathbf{y}, \mathbf{w}, \mathbf{z}, f|\mathbf{x}) &= p(\mathbf{z})p(f)p(\mathbf{w}|f, \mathbf{z})p(\mathbf{y}|\mathbf{w}, \mathbf{x}) \\ &= p(\mathbf{z})p(f)\prod_{n=1}^{N}\left[p(\mathbf{w}_n|f, \mathbf{C}_w(\mathbf{z}))p(\mathbf{y}_n|\mathbf{w}, \mathbf{x}_n)\right]. \end{aligned}$$

We first partition the space $\mathcal{Z}$ of inputs to the function f into a finite set of M variables called inducing inputs $\mathbf{z_u}$ and the remaining inputs, $\mathcal{Z} = \{\mathbf{x_u}, \mathcal{Z}_{\neq \mathbf{x_u}}\}$. The function f is partitioned identically, $f = \{\mathbf{u}, f_{\neq \mathbf{u}}\}$, where $\mathbf{u} = f(\mathbf{x_u})$. We can then rewrite the GP prior as follows, $p(f) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \mathbf{z_u})p(\mathbf{u}|\mathbf{z_u})$.[1] The inducing inputs and outputs, $\{\mathbf{x_u}, \mathbf{u}\}$, will be used to parameterize the approximation. In particular, a variational approximation is judiciously chosen to mirror the form of the joint density:

$$q(\mathbf{w}, \mathbf{z}, f) = q(\mathbf{z})p(f_{\neq \mathbf{u}}|\mathbf{u}, \mathbf{z_u})q(\mathbf{u})p(\mathbf{w}|f, \mathbf{z}), \tag{3}$$

where the variational distribution over $\mathbf{w}$ is made to explicitly depend on remaining variables through the conditional prior, and $q(\mathbf{z})$ is chosen to be a diagonal (mean-field) Gaussian

1. The conditioning on $\mathcal{Z}_{\neq \mathbf{x_u}}$ in $p(f_{\neq \mathbf{u}}|\mathbf{u}, \mathbf{z_u})$ is made implicit here and in the rest of this paper.

densitie, $q(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\mathbf{z}, \text{diag}(\boldsymbol{\sigma}_\mathbf{z}^2))$, and $q(\mathbf{u})$ is chosen to be a correlated multivariate Gaussian, $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \boldsymbol{\mu}_\mathbf{u}, \Sigma_\mathbf{u})$. This approximation allows convenient cancellations yielding a tractable variational lower bound as follows,

$$\begin{aligned}\mathcal{L}_{\text{metaGP}}(\cdot) &= \int_{\mathbf{w},\mathbf{z},f} q(\mathbf{w},\mathbf{z},f) \log \frac{p(\mathbf{z})\cancel{p(f_{\neq \mathbf{u}}|\mathbf{u},\mathbf{z}_\mathbf{u})}p(\mathbf{u}|\mathbf{z}_\mathbf{u})\cancel{p(\mathbf{w}|f,\mathbf{z})}p(\mathbf{y}|\mathbf{w},\mathbf{x})}{q(\mathbf{z})\cancel{p(f_{\neq \mathbf{u}}|\mathbf{u},\mathbf{z}_\mathbf{u})}q(\mathbf{u})\cancel{p(\mathbf{w}|f,\mathbf{z})}} \\ &\approx -\text{KL}[q(\mathbf{z})||p(\mathbf{z})] - \text{KL}[q(\mathbf{u})||p(\mathbf{u}|\mathbf{x}_\mathbf{u})] \\ &\quad + \frac{1}{K}\sum_{k=1}^{K} \int_{\mathbf{w},f} q(\mathbf{w},f|\mathbf{z}_k) \log p(\mathbf{y}|\mathbf{w},\mathbf{x})\end{aligned}$$

where the last expectation has been partly approximated using simple Monte Carlo with the reparameterization trick, i.e. $\mathbf{z}_k \sim q(\mathbf{z})$. We will next discuss how to approximate the expectation $\mathcal{F}_k = \int_{\mathbf{w},f} q(\mathbf{w},f|\mathbf{z}_k) \log p(\mathbf{y}|\mathbf{w},\mathbf{x})$. Note that we split f into $f_{\neq \mathbf{u}}$ and $\mathbf{u}$, and that we can integrate $f_{\neq \mathbf{u}}$ out exactly to give, $q(\mathbf{w}|\mathbf{z}_k,\mathbf{u}) = \mathcal{N}(\mathbf{w}; \mathbf{A}^{(k)}\mathbf{u}, \mathbf{B}^{(k)})$,

$$\mathbf{A}^{(k)} = \mathbf{K}^{(k)}_{f_{\neq \mathbf{u}}\mathbf{u}}\mathbf{K}^{-1}_{\mathbf{u}\mathbf{u}}; \qquad \mathbf{B}^{(k)} = \mathbf{K}^{(k)}_{l,f_{\neq \mathbf{u}}f_{\neq \mathbf{u}}} - \mathbf{K}^{(k)}_{f_{\neq \mathbf{u}}\mathbf{u}}\mathbf{K}^{-1}_{\mathbf{u}\mathbf{u}}\mathbf{K}^{(k)}_{\mathbf{u}f_{\neq \mathbf{u}}} + \sigma_w^2\mathbf{I}. \tag{4}$$

At this point, we can either (i) sample $\mathbf{u}$ from $q(\mathbf{u})$, or (ii) integrate $\mathbf{u}$ out analytically. We opt for the second approach, which gives

$$q(\mathbf{w}|\mathbf{z}_k) = \mathcal{N}(\mathbf{w}; \mathbf{A}^{(k)}\boldsymbol{\mu}_u, \mathbf{B}^{(k)} + \mathbf{A}^{(k)}\Sigma_\mathbf{u}\mathbf{A}^{\intercal,(k)}). \tag{5}$$

In contrast to GP regression and classification in which the likelihood term is factorized point-wise w.r.t. the parameters and thus their expectations only involve a low dimensional integral, we have to integrate out $\mathbf{w}$ in this case, which is of much higher dimensions. When necessary or practical, we resort to Kronecker factored models or make an additional diagonal approximation as follows,

$$\hat{q}(\mathbf{w}|\mathbf{z}_k) = \mathcal{N}(\mathbf{w}; \mathbf{A}^{(k)}\boldsymbol{\mu}_u, \text{diag}(\mathbf{B}^{(k)} + \mathbf{A}^{(k)}\Sigma_\mathbf{u}\mathbf{A}^{\intercal,(k)})). \tag{6}$$

Whilst the diagonal approximation above might look poor from the first glance, it is conditioned on a sample of the latent variables $\mathbf{z}_k$ and thus the weights' correlations are retained after integrating out $\mathbf{z}$. Such correlation is illustrated in 4 where we show the marginal and conditional covariance structures for the weights of a small neural network, separated into diagonal and full covariance models. The diagonal approximation above has been observed to give pathological behaviours in the GP regression case (Bauer et al., 2016), but we did not observe these in practice. $\mathcal{F}_k$ is approximated by $\mathcal{F}_k \approx \int_\mathbf{w} \hat{q}(\mathbf{w}|\mathbf{z}_k) \log p(\mathbf{y}|\mathbf{w},\mathbf{x})$ which can be subsequently efficiently estimated using the *local reparameterization trick* (Kingma et al., 2015). The final lower bound is then optimized to obtain the variational parameterers of $q(\mathbf{u})$, $q(\mathbf{z})$, and estimates for the noise in the meta-GP model, the kernel hyper-parameters and the inducing inputs.

Appendix B. Extra experimental results

B.1. Covariance structures

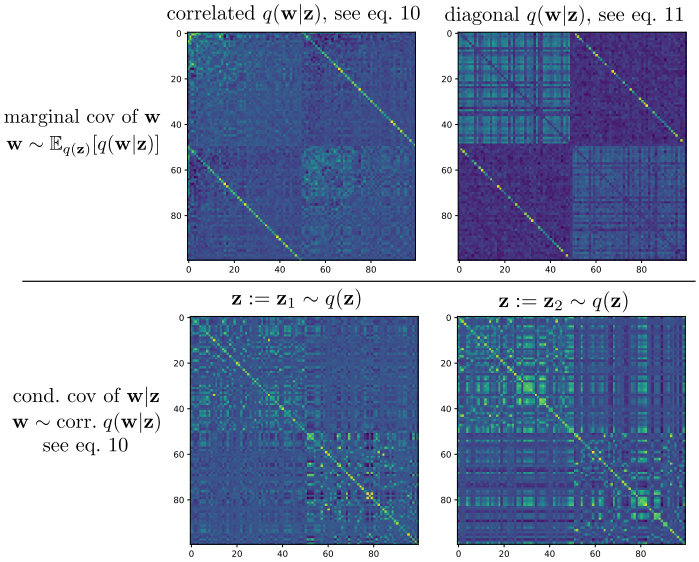

Figure 4: Marginal and conditional covariance structures over weights in a 1x50x1 neural network. Sampling from the posterior of the hierarchical model reveals that even a diagonal GP approximation can capture off-diagonal correlations induced through unit correlations. Also note the off-diagonal bands in the marginal plots above, which indicate the correlation structures induced by the latent variables of the hidden units connecting the layers.We remove the diagonal in the marginal plots for clarity.

B.2. Active learning

We next stress-test the performance of the proposed model in a pool-based active learning setting for real-valued regression, where limited training data is provided initially and the target is to sequentially select points from a pool set to add to the training set. The criterion to select the next best point from the pool set is based on the entropy of the predictive distribution, i.e. we pick one with the highest entropy. In practice, we approximate the predictive density by a Gaussian density, which results in a tractable entropy computation. Note that this selection procedure can be interpreted as selecting points that maximally reduce the posterior entropy of the network parameters Houlsby et al. (2011). Four UCI regression datasets were considered, where each with 40 random train/test/pool splits. For each split, the initial train set has 20 data points, the test set has 100 data points, and the remaining points are used for the pool set, similar to the active learning set-up in Hernández-Lobato and Adams (2015). We compare the performance of the proposed model and inference scheme to that of Gaussian mean-field variational inference and show the average results in 5. Across all runs, we observe that active learning is superior to random

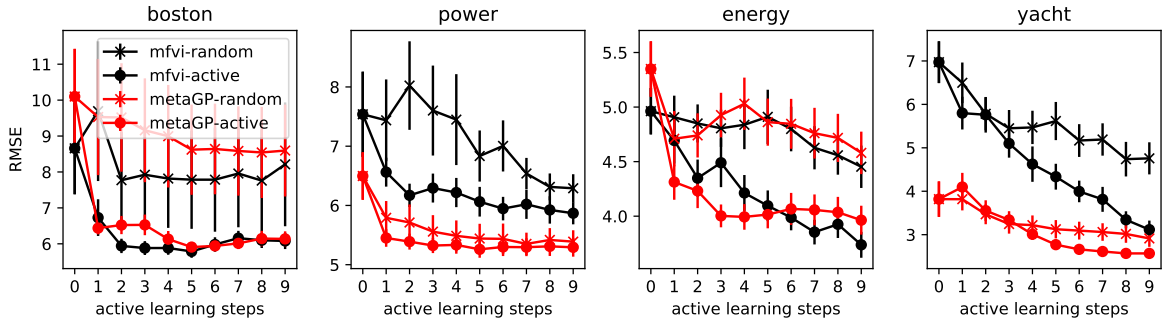

Figure 5: Active learning with BNNs using mean-field Gaussian variational inference [MFVI] and a meta-GP hierarchical prior [MetaGP] on several UCI regression datasets. Each trace shows the root mean squared error (RMSE) on the test set as more data points are selected and moved from the pool set to the training set, averaged over 40 runs. The objective function for selecting points from the pool set is the predictive variance. Best viewed in colour.

selection and more crucially using the proposed model and inference scheme seems to yield comparable or better predictive errors with a similar number of queries. This simple setting quantitatively reveals the inferior performance of MFVI, compared to MAP and metaGP.