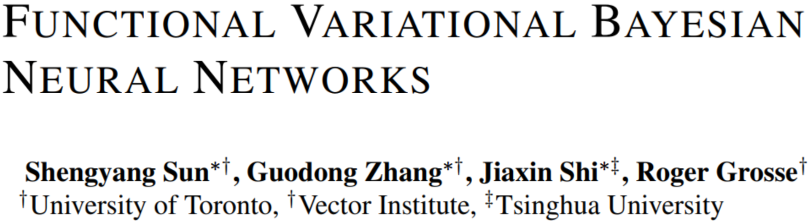


FUNCTIONAL VARIATIONAL BAYESIAN NEURAL NETWORKS

Shengyang Sun[*†], **Guodong Zhang**[*†], **Jiaxin Shi**[*‡], **Roger Grosse**[†]
[†]University of Toronto, [†]Vector Institute, [‡]Tsinghua University

Presented for STA4273 by Ian Shi and Junhao Zhu

Introduction

Main Idea: Define a stochastic process prior on Bayesian Neural Networks, as opposed to weight space prior.

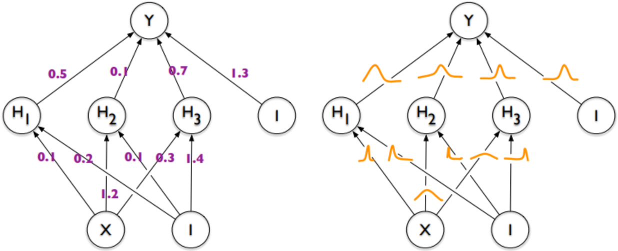
Key Contributions:

- The functional ELBO (fELBO)
- KL Divergence between two stochastic processes
- Techniques for computing fELBO gradients

Background: BNNs

Bayesian Neural Networks (BNNs) introduce a prior on weights $p(\mathbf{w})$.

- Improves performance, as BNNs act as an ensemble of networks
- Allows better quantification of uncertainty compared to regular NNs



Source: Blundell et al. 2015

Problem: Exact inference on weights intractable

Background: Bayes by Backprop

"Weight Uncertainty in Neural Networks" (Blundell et al. 2015)

- Model distribution of each BNN weight using $\mathcal{N}(\mu_i, \sigma_i)$
- Sample realization of BNN from weight posterior $q(\mathbf{w}|\phi)$
- Use reparameterization trick to allow backpropagation
- Train by minimizing ELBO:

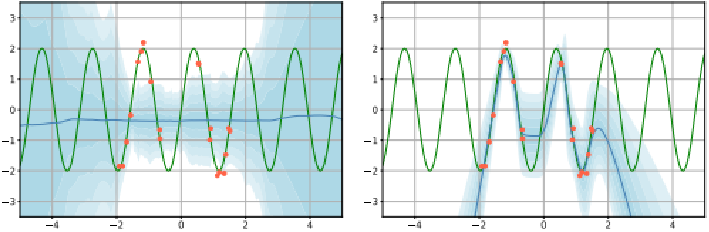
$$\mathcal{L}_q = \mathbb{E}_q[\log p(\mathcal{D}|\mathbf{w})] - \mathsf{KL}[q(\mathbf{w}) \parallel p(\mathbf{w})] \tag{1}$$

  where the expectation is computed by Monte Carlo.

Many related methods (Goan and Fookes 2020), but all place prior over BNN weights, instead of distribution of functions.

Background: Bayes by Backprop

Problem: Results aren't great!



(a) BBB-1 (b) BBB-0.001

Results of Bayes-By-Backprop (BBB) on toy problem. Source: Sun et al. 2019

Functional BNN

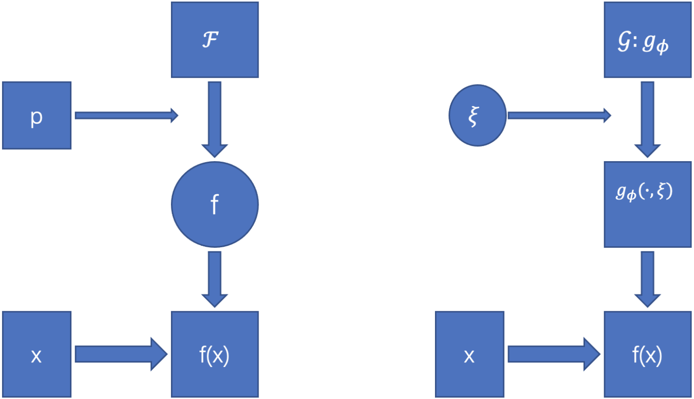


Figure: Idea of fBNN, e.g. $\mathcal{F} = \{f(x) = wx + b : w \in \mathbb{R}, b \in \mathbb{R}\}$

Functional ELBO

Sun et al. introduce the Functional ELBO (fELBO):

$$\mathcal{L}_q = \mathbb{E}_q[\log p(\mathcal{D}|f)] - \mathsf{KL}[q \parallel p] \tag{2}$$

- q is the fBNN posterior $q(f|\phi)$ allowing reparametrization.
- f is a sample from the fBNN posterior $f \sim q$
- p is a prior on a function space.
- $\{f(x) : x \in \mathcal{X}\}$ can be viewed as a stochastic process.

Breakdown:

- Likelihood term $\mathbb{E}_q[\log p(\mathcal{D}|f)]$ computed as in BBB.
- How do we compute $\mathsf{KL}[q \parallel p]$?

Functional ELBO

Given two stochastic processes P and Q:

$$\mathsf{KL}[P \parallel Q] = \sup_{n \in \mathbb{N}, \mathbf{X} \in \mathcal{X}^n} \mathsf{KL}[P_{\mathbf{X}} \parallel Q_{\mathbf{X}}] \tag{3}$$

- $\mathbf{X}$ is the measurement set: a finite set of points where function is evaluated
- $P_{\mathbf{X}}$ is the marginal distribution of functional values at $\mathbf{X}$

"Supremum of marginal KL divergences over all finite sets of inputs"

See Appendix A of Sun et al. 2019 for full proof.

Functional ELBO

Thus:

$$\begin{aligned}
\mathcal{L}(q) &= \mathbb{E}_q[\log p(\mathcal{D}|f)] - \sup_{n\in\mathbb{N},\mathbf{X}\in\mathcal{X}^n} \mathsf{KL}[q(\mathbf{f}^X) \parallel p(\mathbf{f}^X)] && (4)\\
&= \inf_{n\in\mathbb{N},\mathbf{X}\in\mathcal{X}^n} \sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}} \mathbb{E}_q[\log p(y_i|f(\mathbf{x}_i)] - \mathsf{KL}[q(\mathbf{f}^X) \parallel p(\mathbf{f}^X)] && (5)\\
:&= \inf_{n\in\mathbb{N},\mathbf{X}\in\mathcal{X}^n} \mathcal{L}_{\mathbf{X}}(q) && (6)
\end{aligned}$$

Also define

$$\mathcal{L}_n(q) := \inf_{\mathbf{X}\in\mathcal{X}^n} \mathcal{L}_{\mathbf{X}}(q) \qquad (7)$$

as the fELBO restricted to sets of only n points.

Choosing Measurement Sets: Adversarial Methods

What is the best way to chose the measurement set $\mathbf{X}$?

Adversarial Measurement Sets: Cast fELBO as a two-player zero-sum game. One player chooses the BNN, the other chooses the measurement set. Concurrently optimize:

$$\max_{q \in \mathcal{Q}} \mathcal{L}_m(q) := \max_{q \in \mathcal{Q}} \min_{\mathbf{X} \in \mathcal{X}^m} \mathcal{L}_{\mathbf{X}}(q) \tag{8}$$

Doesn't work that well unfortunately. Tends to chose measurement set overlapping training data, meaning functional prior is ignored for extrapolation.

Choosing Measurement Sets: Sampling

Sampling Based Measurement Sets: Define distribution c from which to draw measurement sets:

$$\max_{q\in\mathcal{Q}} \mathcal{L}(q) := \max_{q\in\mathcal{Q}} \mathbb{E}_{\mathcal{D}_s} \mathbb{E}_{\mathbf{X}^M \sim c}\, \mathcal{L}_{\mathbf{X}^M, \mathbf{X}^{\mathcal{D}_s}}(q) \tag{9}$$

- Sampled measurement set should include both training data $(\mathbf{X}^{\mathcal{D}_s})$ and prediction regions $(\mathbf{X}^M)$.

In experiments, authors sample from rectangle $[x_{\min} - d/2,\, x_{\max} + d/2]$

- $x_{\min}$, $x_{\max}$ are the min / max input values along a dimension
- $d = x_{\max} - x_{\min}$

Sampling-Based Measurement Sets

Including all training input in measurement set allows fELBO to lower bound log marginal likelihood such that:

$$\mathcal{L}_{\mathbf{X}}(q) = \log p(\mathcal{D}) - \mathsf{KL}[q(\mathbf{f}^{\mathbf{X}}) \parallel p(\mathbf{f}^{\mathbf{X}}|\mathcal{D})] \leq \log p(\mathcal{D}) \tag{10}$$

See Appendix B.2 of Sun et al. 2019 for full proof.

KL Divergence Gradients

Remaining challenge: How do we compute gradients for KL term if we don't have explicit $q_\phi(\mathbf{f}^\mathbf{X})$?

The gradient of the KL term $\nabla_\phi \mathsf{KL}[q_\phi(\mathbf{f}^\mathbf{X}) \parallel p(\mathbf{f}^\mathbf{X})]$ is:

$$\mathbb{E}_q[\nabla_\phi \log q_\phi(\mathbf{f}^\mathbf{X})] + \mathbb{E}_\xi[\nabla_\phi \mathbf{f}^\mathbf{X}(\nabla_\mathbf{f} \log q(\mathbf{f}^\mathbf{X}) - \nabla_\mathbf{f} \log p(\mathbf{f}^\mathbf{X}))] \tag{11}$$

- First term goes to zero
- The term $\nabla_\phi \mathbf{f}^\mathbf{X}$ computed via backprop.

We need a gradient estimator for $\nabla_\mathbf{f} \log q(\mathbf{f}^\mathbf{X})$ and $\nabla_\mathbf{f} \log p(\mathbf{f}^\mathbf{X})$, which are both intractable.

Spectral Stein Gradient Estimator

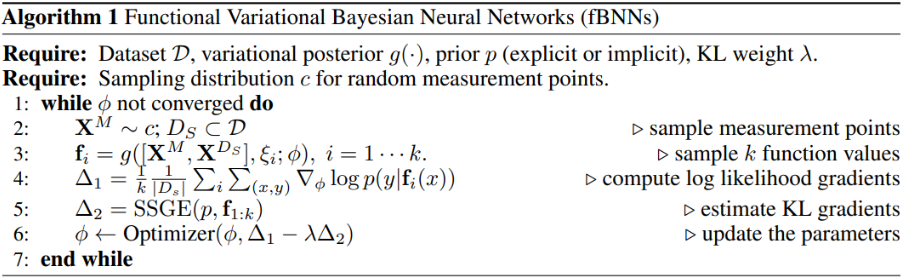
Spectral Stein Gradient Estimator (SSGE) (Shi, Sun, and Zhu 2018) is used to compute gradients for implicit distributions, where the distribution is intractable, but sampling is tractable. Sounds applicable!

The SSGE is given by:

$$\nabla_{x_i} \log q(\mathbf{x}) = g_i(x) = \sum_{j=1}^{\infty} \beta_{ij} \psi_j(\mathbf{x}) \approx \sum_{j=1}^{J} \hat{\beta}_{ij} \hat{\psi}_j(\mathbf{x}) \tag{12}$$

where eigenvalues (β_{ij}) and eigenfunctions (ψ_j) are computed by Nyström approximation. (Williams and Seeger 2001)

Algorithm



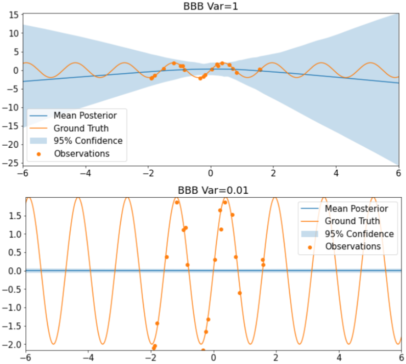
Algorithm 1 Functional Variational Bayesian Neural Networks (fBNNs)

```
Require: Dataset $\mathcal{D}$, variational posterior $g(\cdot)$, prior $p$ (explicit or implicit), KL weight $\lambda$.
Require: Sampling distribution $c$ for random measurement points.
while $\phi$ not converged do
    $\mathbf{X}^M \sim c; D_S \subset \mathcal{D}$    ▷ sample measurement points
    $\mathbf{f}_i = g([\mathbf{X}^M, \mathbf{X}^{D_S}], \xi_i; \phi),\ i = 1 \cdots k.$    ▷ sample $k$ function values
    $\Delta_1 = \frac{1}{k}\frac{1}{|D_s|}\sum_i \sum_{(x,y)} \nabla_\phi \log p(y|\mathbf{f}_i(x))$    ▷ compute log likelihood gradients
    $\Delta_2 = \mathrm{SSGE}(p, \mathbf{f}_{1:k})$    ▷ estimate KL gradients
    $\phi \leftarrow \mathrm{Optimizer}(\phi, \Delta_1 - \lambda\Delta_2)$    ▷ update the parameters
end while
```

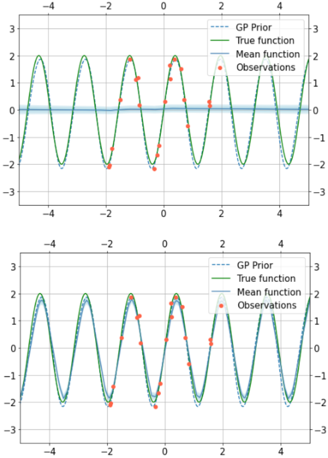
Source: Sun et al. 2019

Experiments (Code Notebook)

Recreating Toy Experiment from Paper:



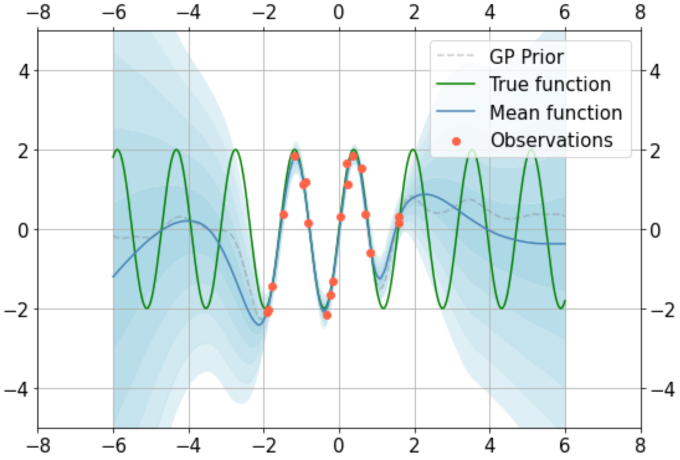
(a) BBB



(b) fBNN with GP Prior (Periodic + RBF kernel)

Experiments (Code Notebook)

Control over fBNN properties possible through GP prior.



Experiments

Authors validate their method on:

- UCI Regression Data Sets
- Contextual Bandits

Functional BNNs compared mainly against BBB on test MSE prediction of held out points. Generally outperforms BBB, but metrics quite close.

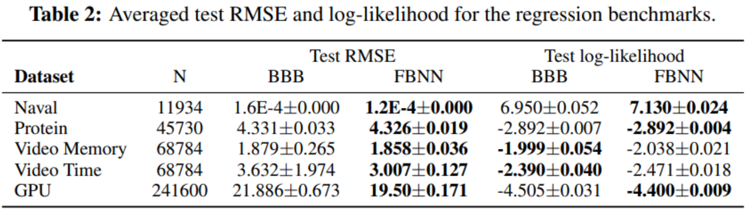


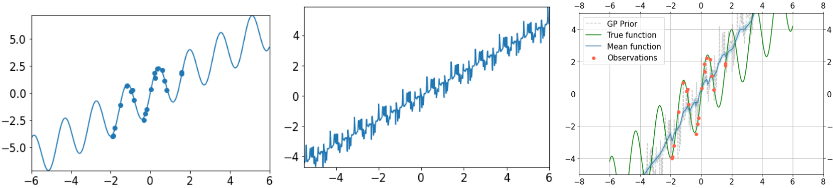
Table 2: Averaged test RMSE and log-likelihood for the regression benchmarks.

		Test RMSE		Test log-likelihood	
Dataset	N	BBB	FBNN	BBB	FBNN
Naval	11934	1.6E-4±0.000	**1.2E-4±0.000**	6.950±0.052	**7.130±0.024**
Protein	45730	4.331±0.033	**4.326±0.019**	-2.892±0.007	**-2.892±0.004**
Video Memory	68784	1.879±0.265	**1.858±0.036**	**-1.999±0.054**	-2.038±0.021
Video Time	68784	3.632±1.974	**3.007±0.127**	**-2.390±0.040**	-2.471±0.018
GPU	241600	21.886±0.673	**19.50±0.171**	-4.505±0.031	**-4.400±0.009**

Limitations (Code Notebook)

Performance of fBNN heavily tied to accuracy of functional prior.

Experiment on linearly Period toy data. Should be easily represented with Linear + Periodic kernel.

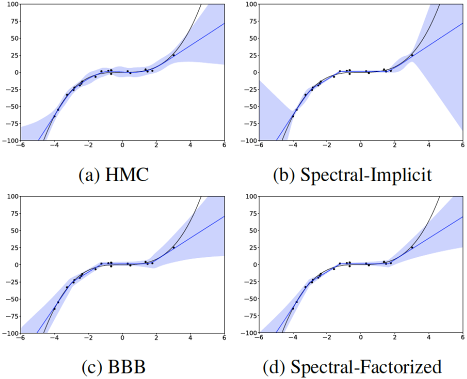


(a) Data (b) Learned Prior (c) fBNN

Have to be careful!

Limitations

SSGE is known to cause underestimates uncertainty in comparison to HMC (Shi, Sun, and Zhu 2018). Shown in figure below.



(a) HMC

(b) Spectral-Implicit

(c) BBB

(d) Spectral-Factorized

Source: Shi, Sun, and Zhu 2018

Conclusion

Key Takeaways: Functional Variational Bayesian Neural Nets

- Function (instead of weight) space prior on BNNs

Functional BNN is trained via the Functional ELBO

- KL divergence between two stochastic processes computed as supremum of marginals on all finite measurement sets.
- Measurement sets used compute KL divergence obtained via sampling
- KL gradients computed using SSGE.

References

Charles Blundell et al. "Weight uncertainty in neural network". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1613–1622.

Ethan Goan and Clinton Fookes. "Bayesian Neural Networks: An Introduction and Survey". In: *Case Studies in Applied Bayesian Data Science*. Springer, 2020, pp. 45–87.

Jiaxin Shi, Shengyang Sun, and Jun Zhu. "A spectral approach to gradient estimation for implicit distributions". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4644–4653.

Shengyang Sun et al. "Functional variational bayesian neural networks". In: *arXiv preprint arXiv:1903.05779* (2019).

Christopher Williams and Matthias Seeger. "Using the Nyström method to speed up kernel machines". In: *Proceedings of the 14th annual conference on neural information processing systems*. CONF. 2001, pp. 682–688.