The Expressiveness of Approximate Inference in Bayesian Neural Networks

Andrew Y. K. Foong*[1], David R. Burt*[1], Yingzhen Li[2], and Richard E. Turner[1]
[1]University of Cambridge, [2]Imperial College London

RIKEN Center, 22nd February 2021

UNIVERSITY OF CAMBRIDGE

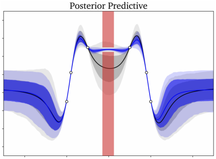
Why Bayesian neural networks?

Bayesian inference allows us to:

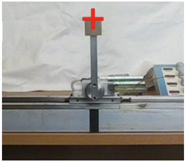
- Represent uncertainty.
- Encode prior beliefs.
- Trade off exploration and exploitation (RL, active learning, BayesOpt).
- Provide framework for continual learning.

BNNs aim to combine benefits of deep learning and Bayesian inference

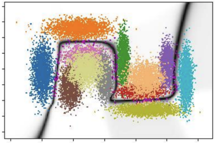
Filos et al. [2019]



Yang et al. [2020]



Deisenroth and Rasmussen [2011]



Pan et al. [2020]

Bayesian neural networks

Probabilistic model:

- Input x, weights θ, neural network f_θ.
- Likelihood $p(\mathcal{D}|\theta) \coloneqq \prod_{n=1}^{N} p(y_n|x_n, \theta) = \prod_{n=1}^{N} p(y_n|f_\theta(x_n))$.
- Prior $p(\theta)$.

Conventional training

Optimise: $\theta_{MAP} = \arg\max_\theta [\log p(\mathcal{D}|\theta) + \log p(\theta)]$.
Predict: $p(y_*|x_*, \theta_{MAP})$.

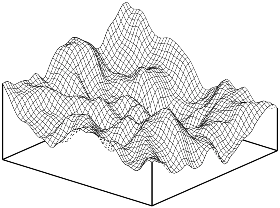
Bayesian inference

Bayes' theorem: $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$.
Predict: $p(y_*|x_*, \mathcal{D}) = \mathbb{E}_{p(\theta|\mathcal{D})} [p(y_*|x_*, \theta)]$.

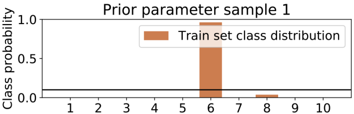
Bayesian approach not without its challenges!

First main challenge — the prior

1. How can we specify a good prior?
   - Model mismatch can lead to poor predictions.
   - Often factorised Gaussian for convenience.
   - Prior sampling can yield insights:



BNN sample from Neal [1995]



Typical prior predictive from Wenzel et al. [2020]

Second main challenge — the posterior

2. How can we perform good inference?
    - Need to approximate high-dimensional integral.
    - Difficult to verify if approximation has succeeded.
    - Is performance due to the model or to the approximation?

These two challenges are linked.

- Often priors are chosen by evaluating the posteriors they induce.
  "**Ye priors shall be known by their posteriors**" [Good, 1983].
- Lack of reliable inference hampers prior evaluation.

This talk will focus on **analysing approximate inference**.

Approximate inference

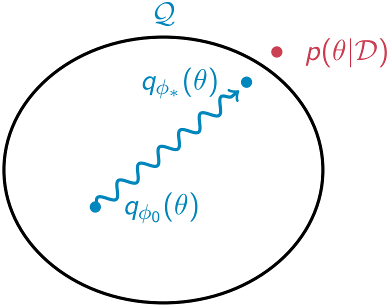
We focus on **approximating family methods**, which assume some tractable parametric form:

$$\mathbb{E}_{p(\theta|\mathcal{D})}\left[p(y_*|\mathrm{x}_*,\theta)\right] \approx \mathbb{E}_{q_\phi(\theta)}\left[p(y_*|\mathrm{x}_*,\theta)\right], \quad q_\phi(\theta) \in \mathcal{Q}.$$

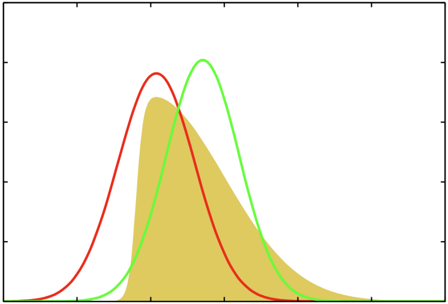
- $\mathcal{Q}$ is the **approximating family**, e.g. set of Gaussian distributions over θ.
- ϕ are parameters, e.g. mean and covariance matrix.
- Approximate inference amounts to choosing ϕ.
- E.g. Laplace approximation, expectation propagation, variational inference (VI).

Variational inference recap

- Choose $q \in \mathcal{Q}$ that minimises $\mathrm{KL}(q_\phi(\theta) \| p(\theta|\mathcal{D}))$.
- In practice optimise ELBO: $\mathbb{E}_{q_\phi(\theta)}[\log p(\mathcal{D}|\theta)] - \mathrm{KL}(q_\phi(\theta) \| p(\theta))$
- Converts integration into optimisation.
- If $p(\theta|\mathcal{D}) \in \mathcal{Q}$, then $q_{\phi_*}(\theta) = p(\theta|\mathcal{D})$.



Examples of approximating family methods



Exact posterior, Laplace, variational inference. From Bishop [2006].

Laplace and VI here share the same Gaussian $\mathcal{Q}$, but choose ϕ differently.

Approximating families

Many choices for $\mathcal{Q}$ available.

- Mean-field/fully-factorised Gaussian $\mathcal{Q}_{MF}$ [Denker and LeCun, 1990, Hinton and Van Camp, 1993]:

$$q_\phi(\theta) = \prod_i \mathcal{N}(\theta_i; \mu_i, \sigma_i^2).$$

- Full-covariance Gaussian $\mathcal{Q}_{FC}$ [MacKay, 1992, Barber and Bishop, 1998]:

$$q_\phi(\theta) = \mathcal{N}(\theta; \mu, \Sigma).$$

- Monte Carlo Dropout, $\mathcal{Q}_{DO}$ [Gal and Ghahramani, 2016].

$$\widehat{\mathsf{W}} = \mathsf{W}\,\mathrm{diag}(\boldsymbol{\epsilon}),$$

  where ϵ is a vector of Bernoulli random variables.

Choosing approximating families

How should we choose the approximating family? This is an old question.

MacKay on **Laplace** with $\mathcal{Q}_{MF}$ vs $\mathcal{Q}_{FC}$:

> *"The diagonal approximation is no good because of the strong posterior correlations in the parameters."* — MacKay [1992]

Hinton & van Camp's response on **VI** with $\mathcal{Q}_{MF}$:

> *"It is not clear how much is lost by ignoring the off-diagonal terms... because in this case the [variational] learning will try to force the noise in the weights to be independent."*
>
> — Hinton and Van Camp [1993]

In modern BNNs $\mathcal{Q}_{MF}$ or $\mathcal{Q}_{DO}$ preferred. Can we justify this choice?

Criteria for success

For an approximating family method to succeed, it must satisfy **two criteria**:

1. The approximating family **must contain good approximations** to the posterior.
2. The method **must then select a good approximate posterior** within this family.

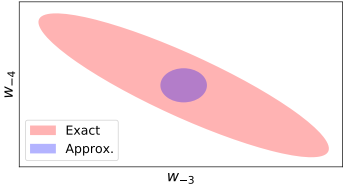
Here, 'good approximation' usually defined in **function space**:

- We often don't care about the weights θ!
- **Interested in predictive** $\mathbb{E}_{p(\theta|\mathcal{D})}\left[p(y_*|x_*, \theta)\right]$.
- Can make assessing impact of approximations less straightforward.

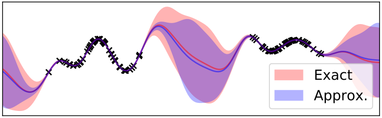
Example: weight space vs function space

Mean-field VI on Bayesian linear regression with RBF features:

$$y(x) = \sum_{i=-10}^{10} w_i \psi_i(x), \quad \psi_i(x) = \exp(-(x-i)^2), \quad w_i \sim \mathcal{N}(0,1)$$



MFVI overconfident in weight space as expected.



But predictions in function space quite accurate! Note “in-between” uncertainty.

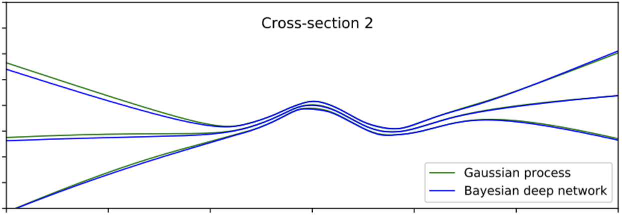
- Weight-space behaviour **doesn’t immediately carry over** to function-space.
- What about for BNNs?

References for the exact posterior

Need good reference to assess inference.

- Exact inference impossible.
- Hamiltonian Monte Carlo possible, but slow, and hard to diagnose.

Deep BNNs approach Gaussian processes as width increases [Matthews et al., 2018, Hron et al., 2020].



3 hidden-layer, width 50 BNN vs. GP. From Matthews et al. [2018].

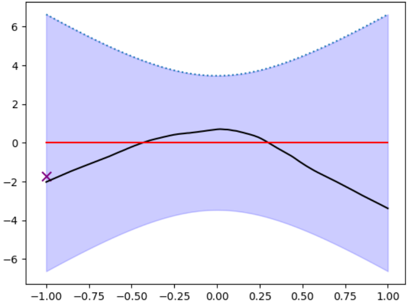
- We use both HMC and GP as references.
- GP expected to be qualitatively suggestive of exact posterior.

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **GP** does:



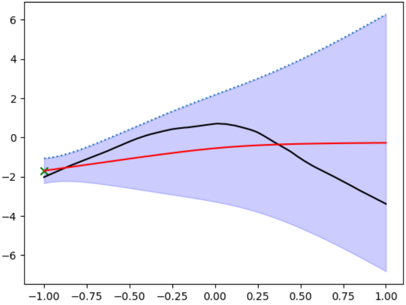
GP BayesOpt using upper confidence bounds: iteration 1

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **GP** does:



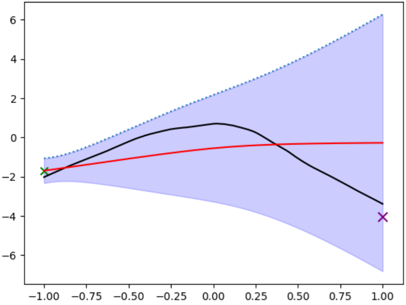
GP BayesOpt using upper confidence bounds: iteration 2

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **GP** does:



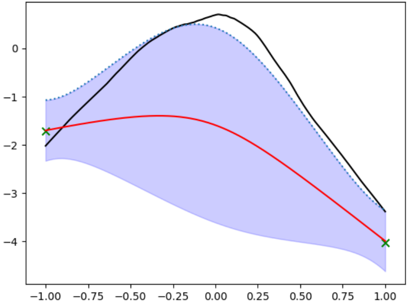
GP BayesOpt using upper confidence bounds: iteration 2

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **GP** does:



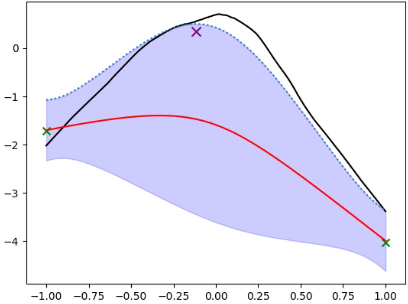
GP BayesOpt using upper confidence bounds: iteration 2

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **GP** does:



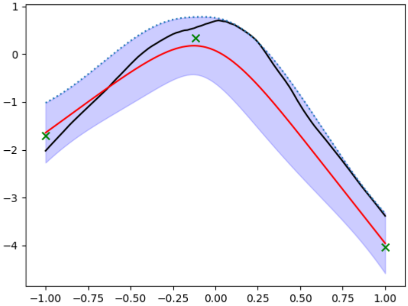
GP BayesOpt using upper confidence bounds: iteration 3

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **GP** does:



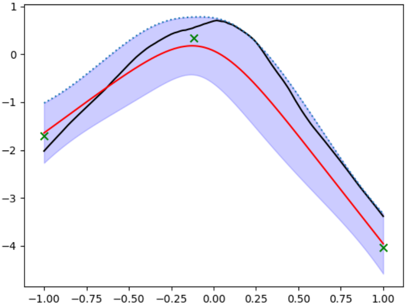
GP BayesOpt using upper confidence bounds: iteration 3

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

GP finds optimum in 3 iterations.



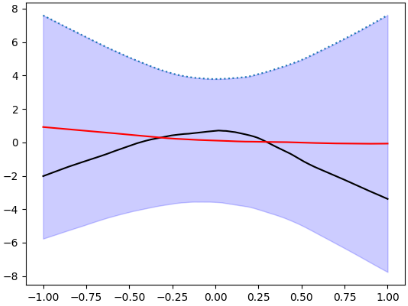
GP BayesOpt using upper confidence bounds: iteration 3

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



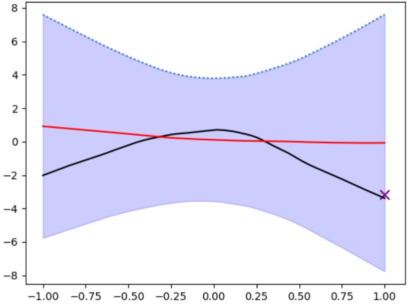
MFVI BayesOpt using upper confidence bounds: iteration 1

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



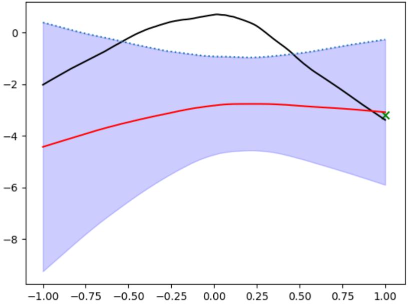
MFVI BayesOpt using upper confidence bounds: iteration 1

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



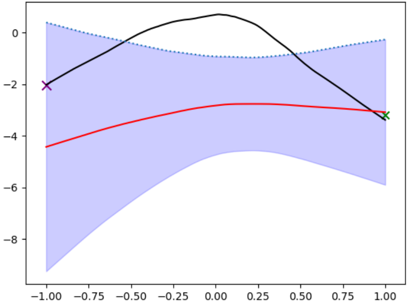
MFVI BayesOpt using upper confidence bounds: iteration 2

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



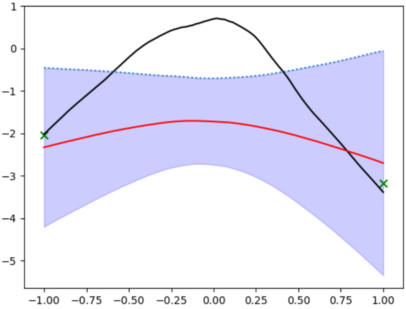
MFVI BayesOpt using upper confidence bounds: iteration 2

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



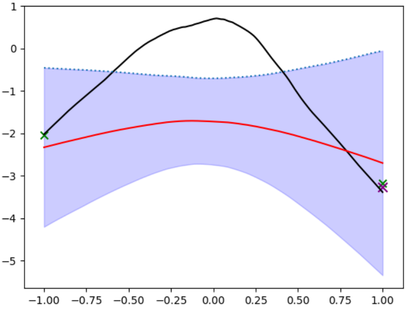
MFVI BayesOpt using upper confidence bounds: iteration 3

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



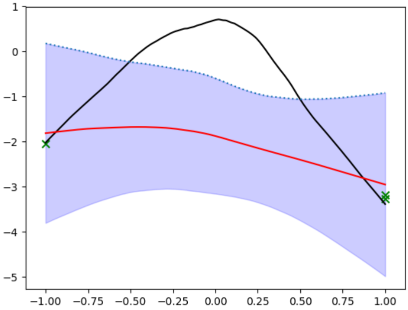
MFVI BayesOpt using upper confidence bounds: iteration 3

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



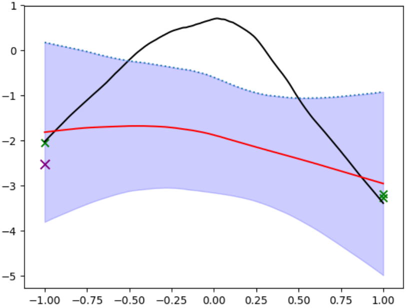
MFVI BayesOpt using upper confidence bounds: iteration 4

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



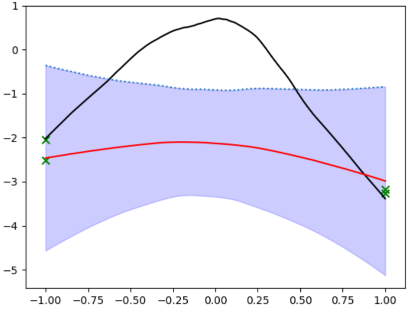
MFVI BayesOpt using upper confidence bounds: iteration 4

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



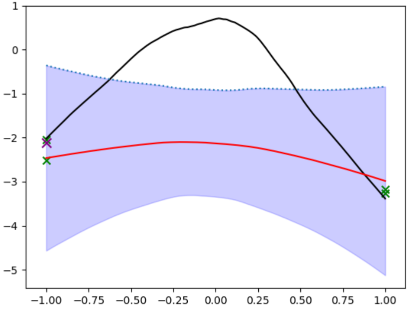
MFVI BayesOpt using upper confidence bounds: iteration 5

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



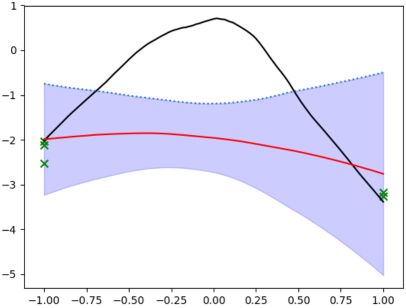
MFVI BayesOpt using upper confidence bounds: iteration 5

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



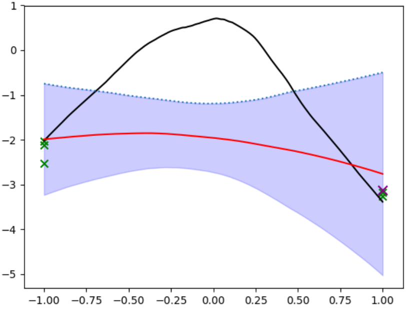
MFVI BayesOpt using upper confidence bounds: iteration 6

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



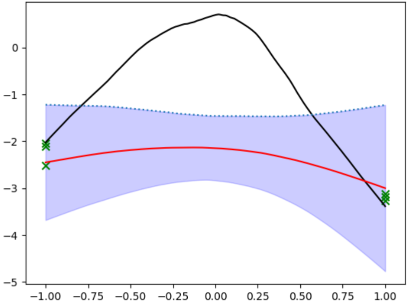
MFVI BayesOpt using upper confidence bounds: iteration 6

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



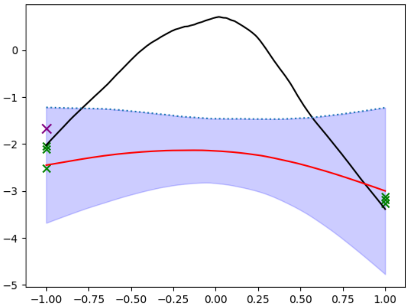
MFVI BayesOpt using upper confidence bounds: iteration 7

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

Here's how the **MFVI BNN** does:



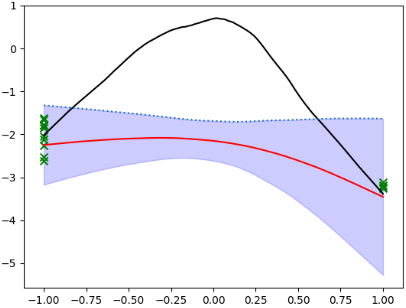
MFVI BayesOpt using upper confidence bounds: iteration 7

How does MFVI compare with GP?

Bayesian optimisation on toy dataset, using

1. single hidden layer MFVI
2. the equivalent infinite-width GP

MFVI still can't find optimum after 15 iterations! Why?



MFVI BayesOpt using upper confidence bounds: iteration 15

1HL Dropout BNNs $\rightarrow$ convex predictive variance

Let $\mathbb{V}[f(x)] := \mathbb{E}[(f_\theta(x) - \mathbb{E}[f_\theta(x)])^2]$ be **predictive variance at** x.

Theorem 1 (F., B., Li & Turner 2020).
For any single hidden layer network with ReLU nonlinearities and a distribution of weights in $\mathcal{Q}_{DO}$, if dropout is not applied to the input layer, $\mathbb{V}[f(x)]$ is convex in x.

- 1HL dropout networks with ReLU activations **can't have in-between uncertainty!**
- A weaker statement is true if input layer is also dropped out.

Proof sketch of theorem 1

Dropout applied independently to each neuron, so:

$$\mathbb{V}[f(x)] = \mathbb{V}\left[\sum_{i=1}^{H} w_i \phi\left(a_i(x)\right) + b\right] \quad (1)$$

$$= \sum_{i=1}^{H} \mathbb{V}\left[w_i \phi\left(a_i(x)\right)\right] + \mathbb{V}[b] \quad (2)$$

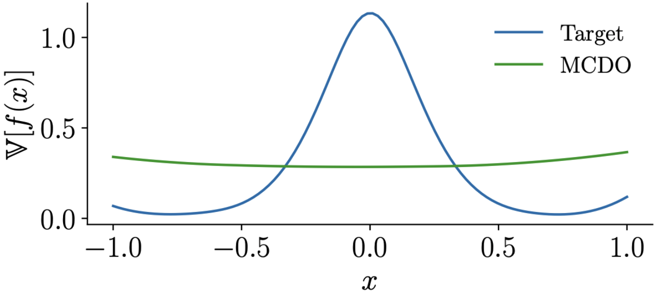
- As the input weights are deterministic,

$$\mathbb{V}\left[w_i \phi\left(a_i(x)\right)\right] = \mathbb{V}\left[w_i\right] \phi\left(a_i(x)\right)^2$$

- $a_i(x)$ is an affine function of x, and ϕ^2 is convex, so $\phi\left(a_i(x)\right)^2$ is convex in x.
- $\mathbb{V}[f(x)]$ is a positive linear combination of convex functions!

Numerical verification of theorem 1

- Obtain reference predictive variance function from a GP.
- Perform gradient descent to **directly minimise** $(\mathbb{V}_{\text{dropout}}[f(x)] - \mathbb{V}_{\text{target}}[f(x)])^2$ on a grid.



MC dropout predictive variance can't match target variance **even when explicitly trained to**, due to theorem 1.

What about mean-field $\mathcal{Q}_{MF}$?

- In dropout proof, we used that the bottom layer was deterministic.
- Does a similar result hold for mean-field Gaussian $\mathcal{Q}_{MF}$, where bottom layer is stochastic?

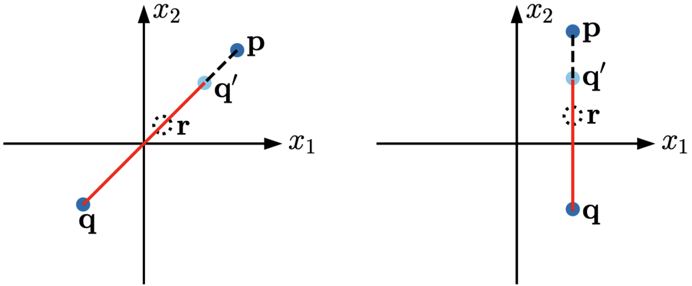
Theorem 2 (F., B., Li & Turner 2020).

There exist line segments in input space, $\overrightarrow{pq}$, such that for any single hidden layer network with ReLU nonlinearities and a distribution of weights in $\mathcal{Q}_{MF}$, for all $r \in \overrightarrow{pq}$,

$$\mathbb{V}[f(r)] \leq \mathbb{V}[f(p)] + \mathbb{V}[f(q)].$$

Constraint is weaker than convexity in theorem 1, but still implies a lack of in-between uncertainty!

Line segments of bounded variance

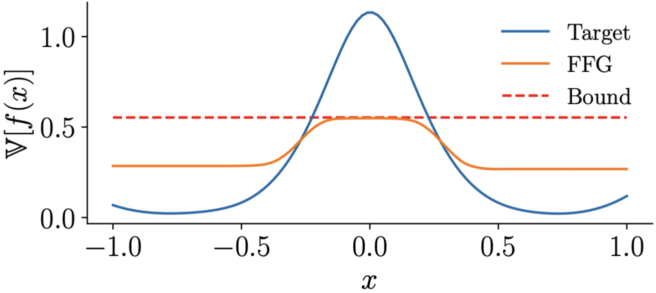


2 example line segments in BNN input space where theorem 2 applies.

- $\mathbb{V}[f(r)] \leq \mathbb{V}[f(p)] + \mathbb{V}[f(p)]$ on the red line segment.
- If input is 1-dimensional, applies to any line segment crossing origin.
- Empirically find in-between uncertainly lacking on *random* line segments.
- Could be symptomatic of more general pathologies.

Numerical verification of theorem 2

- Obtain reference predictive variance function from a GP.
- Perform gradient descent to **directly minimise** $(\mathbb{V}_{\text{mean-field}}[f(x)] - \mathbb{V}_{\text{target}}[f(x)])^2$ on a grid.



Fully-factorised Gaussian (FFG) BNN predictive variance can't match target variance **even when explicitly trained to**, due to theorem 2.

Intuition for theorem 2

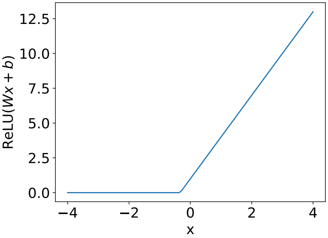
Proof more involved than dropout case.

- Single hidden layer NNs are universal function approximators.
- Surprising that variance of a mean-field BNN is *not* universal!

Intuition:

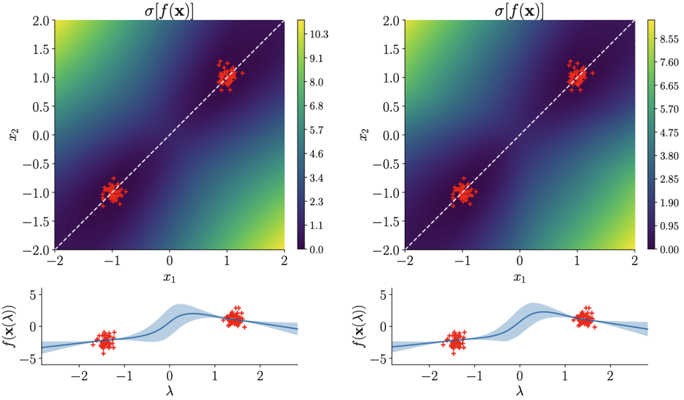
$$\text{Mean field} \implies \text{Variance of sum} = \text{Sum of variances}$$

But variance of each neuron is half bowl shaped:



So variance of any sum is approximately bowl-shaped.

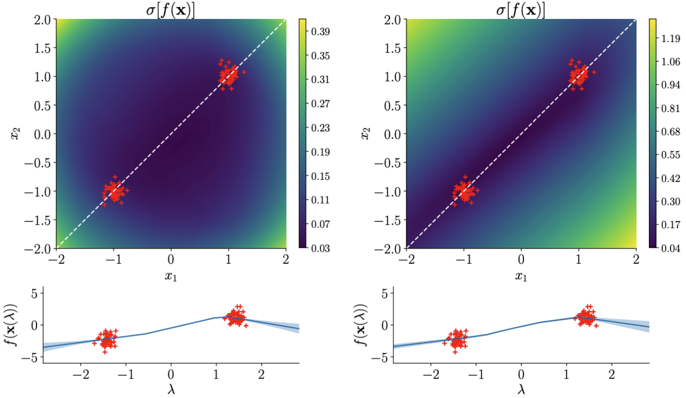
What about an actual inference task?



(a) Infinite-width limit GP (b) HMC

References for exact predictive show plenty of in-between uncertainty.

What about an actual inference task?



(c) MFVI (d) MCDO

- VI with $\mathcal{Q}_{MF}$ or $\mathcal{Q}_{DO}$ loses in-between uncertainty.
- In this case, approximate inference, rather than the model, responsible for overconfidence!

Back to the criteria

1. The approximating family **must contain good approximations** to the posterior. ✗
2. The method **must then select a good approximate posterior** within this family.

If in-between uncertainty desired, **the first criterion is not satisfied** for $\mathcal{Q}_{MF}$ or $\mathcal{Q}_{DO}$ with one hidden layer.

Hence *cannot* be fixed by:

- Choosing a better prior.
- Using a better optimiser.
- Using a tempered posterior, e.g., Wenzel et al. [2020].
- Minimising a different divergence.
- Etc.

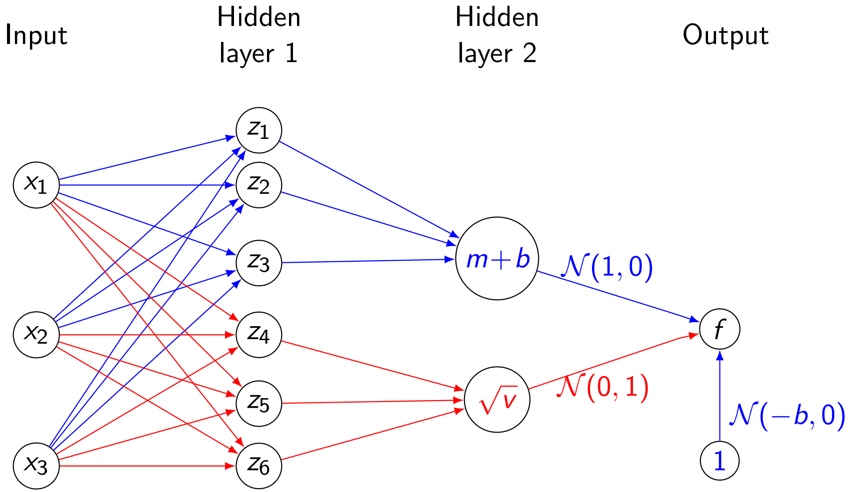
What about **deeper networks**?

Deep networks can have in-between uncertainty

Theorem 3 (F., B., Li & Turner 2020).
Let $A \subset \mathbb{R}^d$ be compact, and $m : A \to \mathbb{R}$, $v : A \to \mathbb{R}_+$ be both continuous. For any $\epsilon > 0$, there exists a sufficiently wide 2HL *ReLU network f, s.t. we can find a distribution in $\mathcal{Q}$ with $\|\mathbb{E}[f] - m\|_\infty < \epsilon$ and $\|\mathbb{V}[f] - v\|_\infty < \epsilon$; where $\mathcal{Q} \in \{\mathcal{Q}_{DO}, \mathcal{Q}_{MF}\}$.*

- Universality theorem for first two moments of marginal of predictive distribution of random networks.
- Just because these networks exist doesn't mean they are easy to find with conventional approximate Bayesian inference (e.g. VI).
- N.B. Only applies to $\mathcal{Q}_{DO}$ if dropout is *not* applied to input layer.

Construction for mean-field $\mathcal{Q}_{MF}$

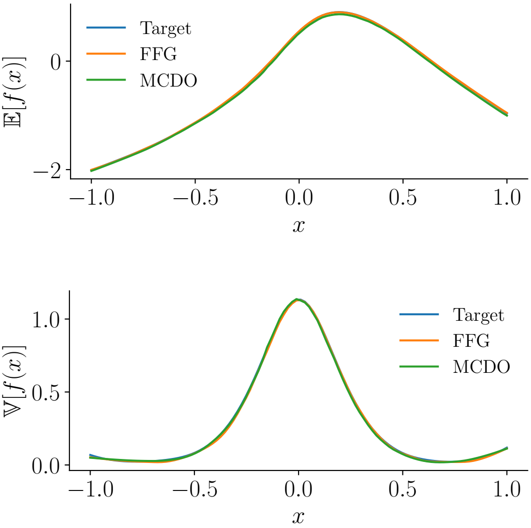


with $b = \min_{x \in A} m(x)$.

So $f \approx 1 \cdot \phi(m+b) + \gamma \cdot \phi(\sqrt{v}) - b \approx m + \gamma\sqrt{v}, \quad \gamma \sim \mathcal{N}(0,1)$.

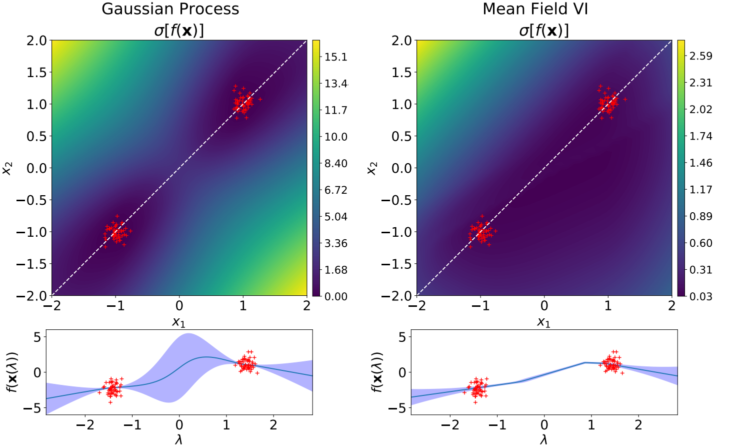
Numerical verification of Theorem 3

Try to fit mean and variance function from before, but with 2HL net:



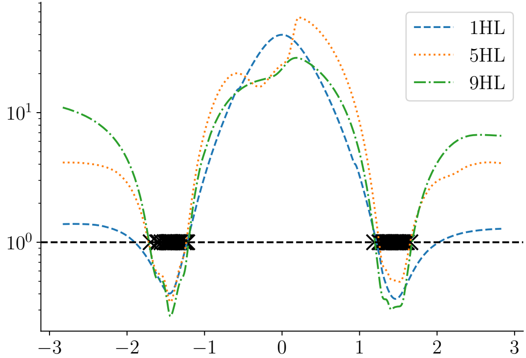
Variational Inference in Deep Nets

Does theorem 3 imply good uncertainty quantification with VI in deep BNNs?



Variational Inference in Deep Nets

Does theorem 3 imply good uncertainty quantification with VI in deep BNNs?

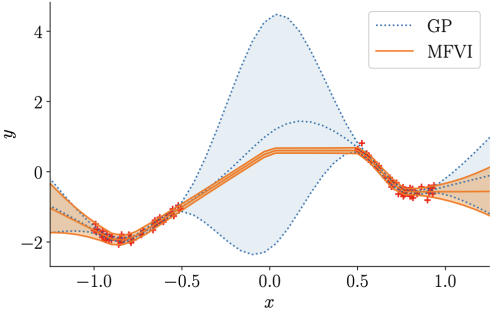


Overconfidence ratio $(\mathbb{V}_{GP}[f]/\mathbb{V}_{MFVI}[f])^{1/2}$ between two clusters of data.

Effect of initialisation

Is this behaviour due to the objective, the optimiser, or something else?

- Initialise 2HL BNN by matching GP mean and variance.
- Then optimise mixture of ELBO and squared error objective.
- Gradually move to just optimising ELBO.



BNN that starts with in-between uncertainty loses it once ELBO optimisation converges!

Limitations of Theorem 3

- Unclear how wide is "sufficiently wide".
- Only tells us about one-dimensional marginal distributions.
- Only tells us about first and second moments.
- Doesn't tell us how to find these 'good' approximate posteriors.

For in-between uncertainty in **VI** in deep BNNs with $\mathcal{Q}_{DO}, \mathcal{Q}_{MF}$:

Criteria for success

1. The approximating family **must contain good approximations** to the posterior. ✓
2. The method **must then select a good approximate posterior** within this family. ✗

Active learning case study

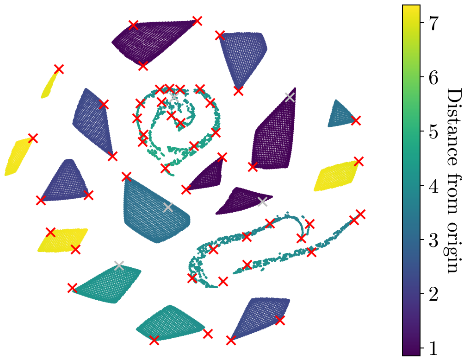
- Goal is to select informative data points to label.
- Common heuristic: Select points with high predictive variance.
- How do issues with uncertainty estimation affect performance?
- We consider a dataset where we observe active learning **fails**.
- Naval regression dataset, $N = 11934$, $D = 14$.

Table 1: Test RMSEs after 50^{th} iteration of active learning.

	1 HL	4 HL
NN-GP Active	0.04 ± 0.00	0.05 ± 0.00
NN-GP Random	0.12 ± 0.01	0.16 ± 0.01
MFVI Active	0.94 ± 0.11	0.31 ± 0.02
MFVI Random	0.15 ± 0.01	0.32 ± 0.01

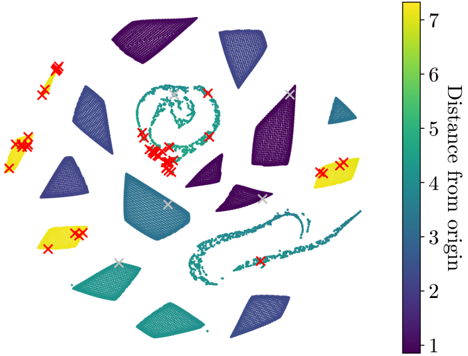
Can in-between uncertainty explain why active learning fails to improve over random for MFVI?

t-SNE plot of 1HL NN-GP acquisitions



- Points chosen at ‘corners’ of clusters.
- Every cluster sampled.

t-SNE plot of 1HL MFVI acquisitions



- ‘Outermost’ clusters favoured.
- ‘In-between’ clusters ignored.
- Effect lessens somewhat in deeper networks, but:
  - Approximate inference still much worse than exact NN-GP.
  - Struggles to outperform random.

Limitations and follow-up work

Limitations:

- Theorems don't explain empirical behaviour of deep BNNs.
- When is in-between uncertainty actually important?
- Focus on regression, not classification.
- *Very* difficult to find **reliable** references for the true posterior in big networks.

Subsequent work (Farquhar et al. [2020]), claims $\mathcal{Q}_{MF}$ less restrictive in deeper nets. However:

- We observe lack of in-between uncertainty in deep nets trained with VI.
- Some conclusions rely on performance of methods on benchmarks such as ImageNet $\neq$ accurate posterior inference.

Conclusions

- Approximate inference with $\mathcal{Q}_{MF}$ and $\mathcal{Q}_{DO}$ in BNNs can lose qualitative features of the exact predictive.
- In 1HL BNNs, in-between uncertainty is provably absent.
- In deeper BNNs, in-between uncertainty is empirically lost.
- In-between uncertainty can mean the difference between outperforming random baseline and not, in active learning.
- Further work is needed to understand exact vs. approximate inference in, e.g. large convolutional networks.

Thanks for listening!

References I

D. Barber and C. M. Bishop. Ensemble learning in bayesian neural networks. *Nato ASI Series F Computer and Systems Sciences*, 168: 215–238, 1998.

C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

M. Deisenroth and C. E. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472. Citeseer, 2011.

J. S. Denker and Y. LeCun. Transforming neural-net output levels to probability distributions. In *Proceedings of the 3rd International Conference on Neural Information Processing Systems*, pages 853–859, 1990.

S. Farquhar, L. Smith, and Y. Gal. Try depth instead of weight correlations: Mean-field is a less restrictive assumption for deeper networks. *arXiv preprint arXiv:2002.03704*, 2020.

References II

A. Filos, S. Farquhar, A. N. Gomez, T. G. Rudner, Z. Kenton, L. Smith, M. Alizadeh, A. de Kroon, and Y. Gal. A systematic comparison of Bayesian deep learning robustness in diabetic retinopathy tasks. *arXiv preprint arXiv:1912.10481*, 2019.

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

I. J. Good. *Good thinking: The foundations of probability and its applications*. U of Minnesota Press, 1983.

G. E. Hinton and D. Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993.

References III

J. Hron, Y. Bahri, R. Novak, J. Pennington, and J. Sohl-Dickstein. Exact posterior distributions of wide Bayesian neural networks. *arXiv preprint arXiv:2006.10541*, 2020.

D. J. MacKay. A practical Bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

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

P. Pan, S. Swaroop, A. Immer, R. Eschenhagen, R. E. Turner, and M. E. Khan. Continual deep learning by functional regularisation of memorable past. *arXiv preprint arXiv:2004.14070*, 2020.

References IV

F. Wenzel, K. Roth, B. Veeling, J. Swiatkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin. How good is the Bayes posterior in deep neural networks really? In *International Conference on Machine Learning*, pages 10248–10259. PMLR, 2020.

W. Yang, L. Lorch, M. A. Graule, H. Lakkaraju, and F. Doshi-Velez. Incorporating interpretable output constraints in Bayesian neural networks. *arXiv preprint arXiv:2010.10969*, 2020.