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

Joint Talk
11th March 2021

UNIVERSITY OF CAMBRIDGE

Challenges for BNNs

1. How can we specify a **good prior**?
    - Cold posterior effect suggests challenges [Wenzel et al., 2020].
    - Renewed interest [Wilson and Izmailov, 2020, Fortuin et al., 2021].
2. How can we perform **good inference**?
    - MCMC and VI don't come with practical guarantees.
    - Is performance due to the Bayesian model or the approximation?

These 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.

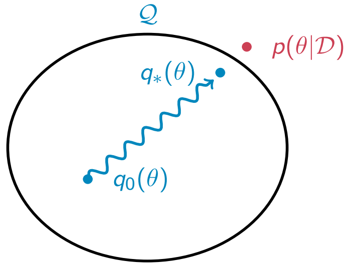
This talk will focus on **analysing approximate inference**.

Approximate inference

We focus on variational methods, which assume some tractable parametric form for approximate posterior:

$$p(y_*|x_*, \mathcal{D}) = \mathbb{E}_{p(\theta|\mathcal{D})}\left[p(y_*|x_*, \theta)\right] \approx \mathbb{E}_{q(\theta)}\left[p(y_*|x_*, \theta)\right], \quad q(\theta) \in \mathcal{Q}.$$

- $p(\theta|\mathcal{D})$ is exact posterior, $q(\theta)$ is approximate posterior.
- $\mathcal{Q}$ is the variational family, e.g. **mean-field (fully-factorised) Gaussian**, or **Monte Carlo dropout**.
- Choose $q \in \mathcal{Q}$ that minimises $\mathrm{KL}(q_\phi(\theta) \| p(\theta|\mathcal{D}))$.



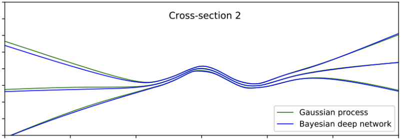
Criteria for success

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

How can we tell if the approximation is good? Need a reference.

- *Very* difficult problem in large models.
- Hamiltonian Monte Carlo possible, but slow, and hard to diagnose.

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



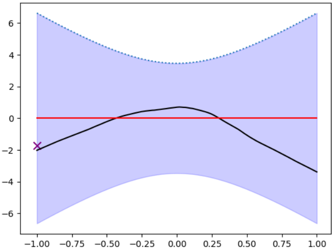
Restrict our study to small datasets, and regression tasks.

How does MFVI compare with NN-GP?

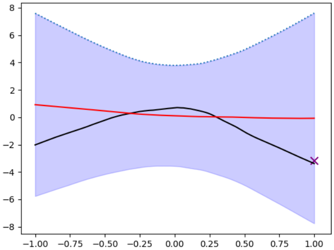
Bayesian optimisation on toy dataset, using

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

GP



MFVI

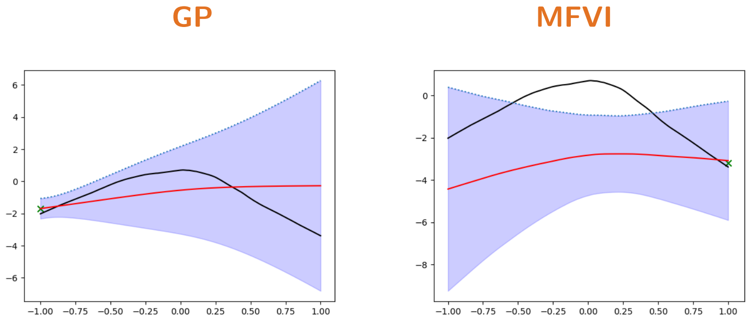


GP versus MFVI BayesOpt using upper confidence bounds: iteration 1

How does MFVI compare with NN-GP?

Bayesian optimisation on toy dataset, using

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



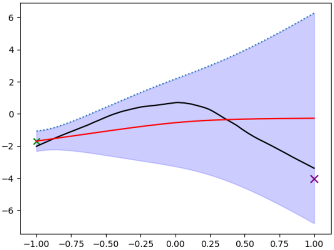
GP versus MFVI BayesOpt using upper confidence bounds: iteration 2

How does MFVI compare with NN-GP?

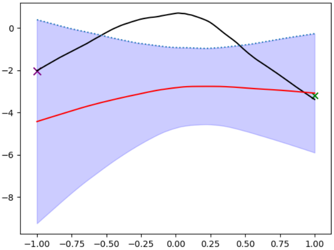
Bayesian optimisation on toy dataset, using

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

GP



MFVI



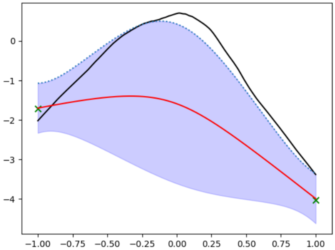
GP versus MFVI BayesOpt using upper confidence bounds: iteration 2

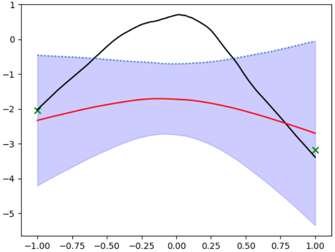
How does MFVI compare with NN-GP?

Bayesian optimisation on toy dataset, using

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

GP **MFVI**





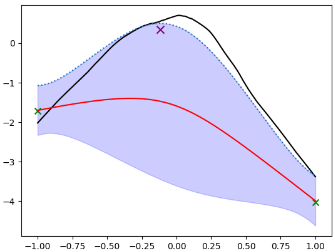
GP versus MFVI BayesOpt using upper confidence bounds: iteration 3

How does MFVI compare with NN-GP?

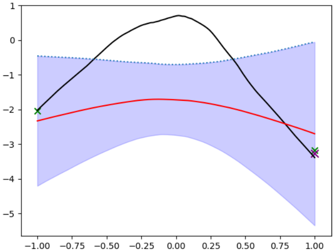
Bayesian optimisation on toy dataset, using

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

GP



MFVI



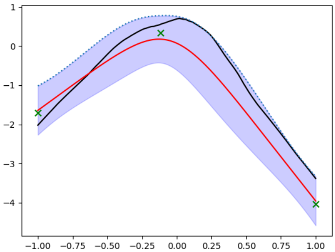
GP versus MFVI BayesOpt using upper confidence bounds: iteration 3

How does MFVI compare with NN-GP?

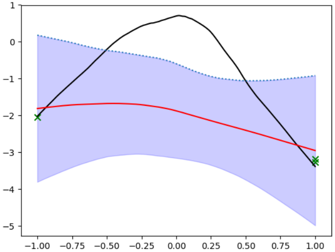
Bayesian optimisation on toy dataset, using

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

GP



MFVI



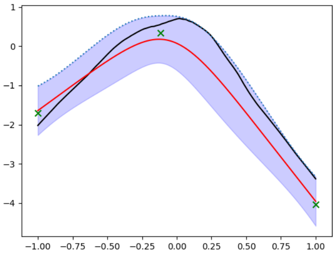
GP versus MFVI BayesOpt using upper confidence bounds: iteration 4

How does MFVI compare with NN-GP?

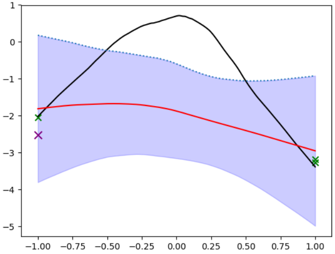
Bayesian optimisation on toy dataset, using

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

GP



MFVI



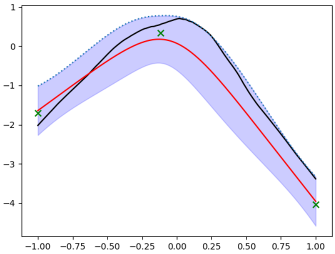
GP versus MFVI BayesOpt using upper confidence bounds: iteration 4

How does MFVI compare with NN-GP?

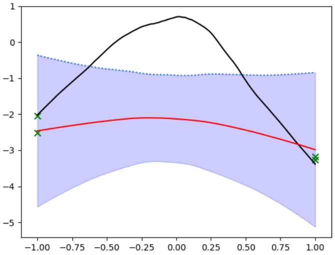
Bayesian optimisation on toy dataset, using

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

GP



MFVI



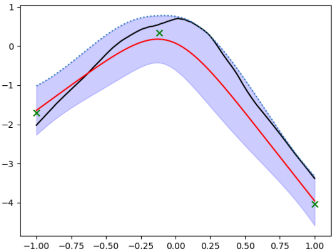
GP versus MFVI BayesOpt using upper confidence bounds: iteration 5

How does MFVI compare with NN-GP?

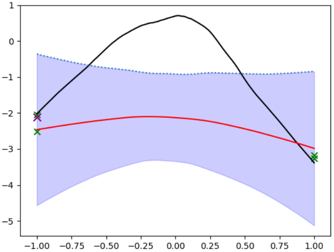
Bayesian optimisation on toy dataset, using

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

GP



MFVI



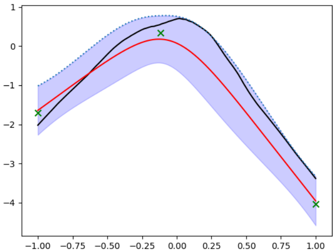
GP versus MFVI BayesOpt using upper confidence bounds: iteration 5

How does MFVI compare with NN-GP?

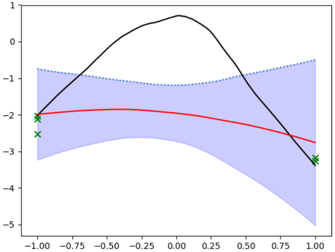
Bayesian optimisation on toy dataset, using

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

GP



MFVI



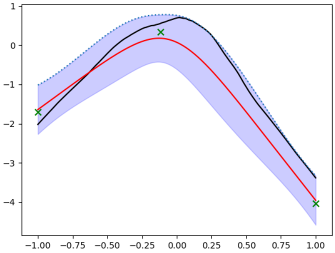
GP versus MFVI BayesOpt using upper confidence bounds: iteration 6

How does MFVI compare with NN-GP?

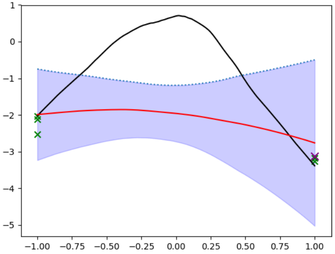
Bayesian optimisation on toy dataset, using

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

GP



MFVI

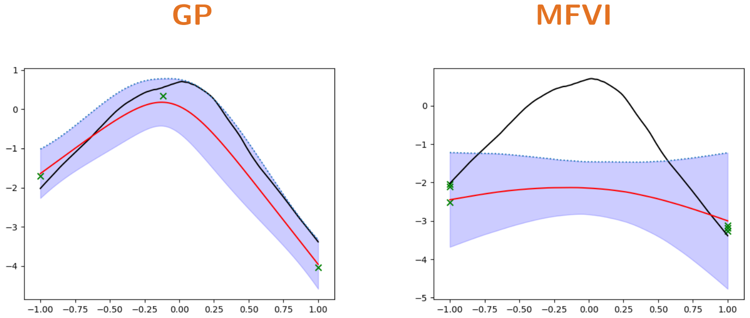


GP versus MFVI BayesOpt using upper confidence bounds: iteration 6

How does MFVI compare with NN-GP?

Bayesian optimisation on toy dataset, using

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



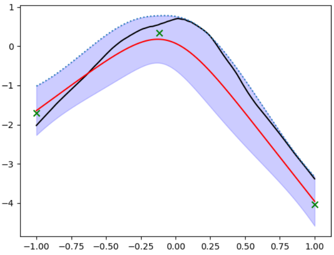
GP versus MFVI BayesOpt using upper confidence bounds: iteration 7

How does MFVI compare with NN-GP?

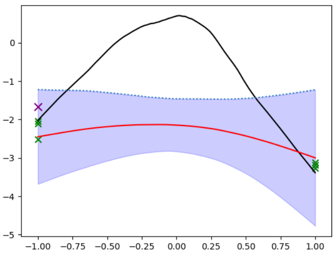
Bayesian optimisation on toy dataset, using

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

GP



MFVI



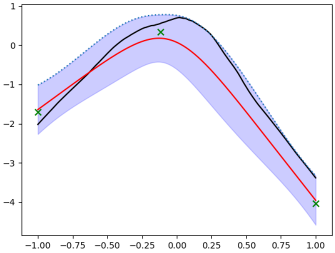
GP versus MFVI BayesOpt using upper confidence bounds: iteration 7

How does MFVI compare with NN-GP?

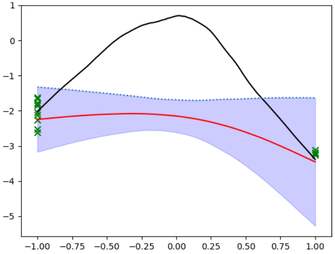
Bayesian optimisation on toy dataset, using

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

GP



MFVI



GP versus MFVI BayesOpt using upper confidence bounds: iteration 15

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

Single hidden layer approximate BNNs

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).
There exist line segments in input space, $\overrightarrow{pq}$, such that for any single hidden layer ReLU network with a mean-field Gaussian weight distribution, for all $r \in \overrightarrow{pq}$,

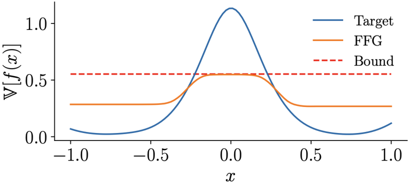
$$\mathbb{V}[f(r)] \leq \mathbb{V}[f(p)] + \mathbb{V}[f(q)].$$

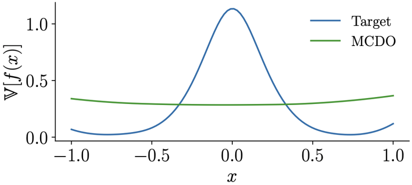
Theorem 2 (F., B., Li & Turner 2020).
For any single hidden layer ReLU network with an MC Dropout weight distribution, if dropout is not applied to the input layer, $\mathbb{V}[f(x)]$ is convex in x.

These 1HL BNNs **can't have in-between uncertainty!**

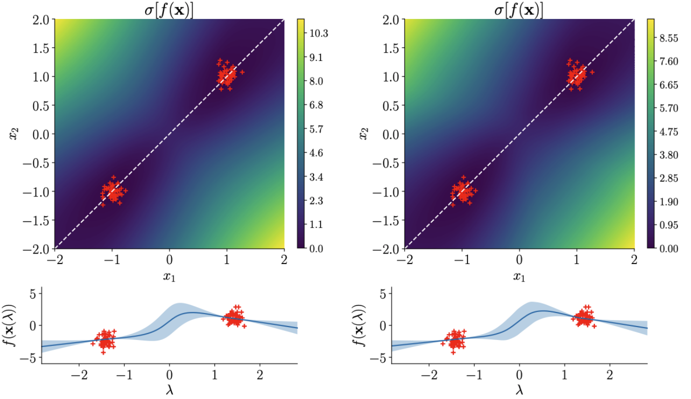
Numerical verification of theorems 1 and 2

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





What about an actual inference task?

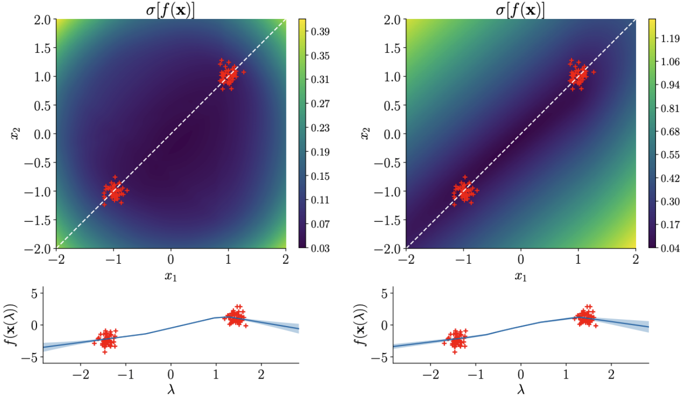


(a) Infinite-width limit GP

(b) HMC

References for exact predictive both show in-between uncertainty.

What about an actual inference task?



(c) MFVI (d) MCDO

- VI loses in-between uncertainty.
- In this case, approximate inference, rather than the model, is provably responsible!

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 mean-field Gaussian or MC Dropout ReLU nets 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 $\mathcal{X} \subset \mathbb{R}^d$ be compact, and $m : \mathcal{X} \to \mathbb{R}$, $v : \mathcal{X} \to \mathbb{R}_+$ be both continuous. For any $\epsilon > 0$, there exists a sufficiently wide 2HL *ReLU network f, with a mean-field Gaussian/MC Dropout distribution satisfying $\|\mathbb{E}[f] - m\|_\infty < \epsilon$ and $\|\mathbb{V}[f] - v\|_\infty < \epsilon$.*

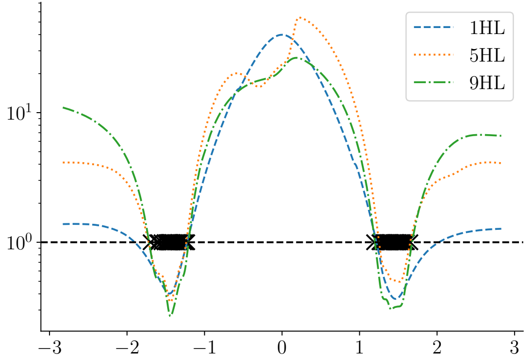
Universality theorem for first two moments of marginal of predictive distribution of random networks.

Criteria for success in deep networks

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

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.

Limitations and conclusions

Limitations:

- References for exact inference difficult in large models.
- Focus on small-scale regression datasets.
- Theorem 3 doesn't explain observed behaviour in deep nets.
- In-between uncertainty isn't everything — good sanity check.

Conclusions:

- Approximate inference with mean-field Gaussian and MC dropout posteriors 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.
- We are still very far from understanding exact vs. approximate inference in, e.g. large convolutional networks.

Thanks for listening!

References I

V. Fortuin, A. Garriga-Alonso, F. Wenzel, G. Rätsch, R. Turner, M. van der Wilk, and L. Aitchison. Bayesian neural network priors revisited. *arXiv preprint arXiv:2102.06571*, 2021.

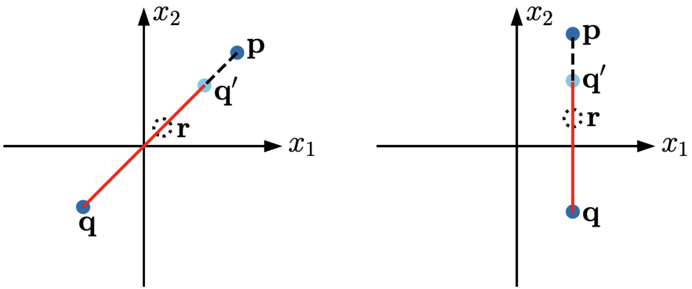
I. J. Good. *Good thinking: The foundations of probability and its applications*. U of Minnesota Press, 1983.

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.

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.

A. G. Wilson and P. Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.

Line segments of bounded variance



2 example line segments in BNN input space where theorem 1 applies.

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

Proof sketch of theorem 2

Dropout applied independently to each neuron, so:

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

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

- 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!

Intuition for theorem 1

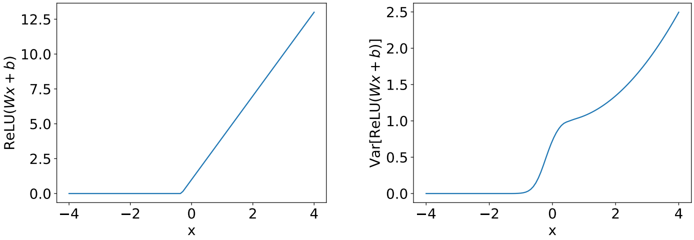
Proof more involved than dropout case.

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

Intuition:

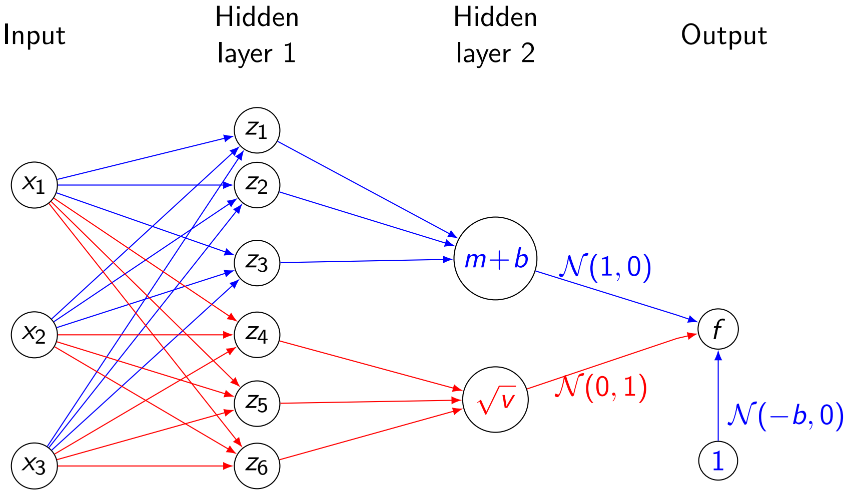
$$\text{Mean field} \implies \text{Variance of sum} = \text{Sum of variances}$$

But variance of each neuron is half bowl shaped:



So variance of any sum is approximately bowl-shaped.

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

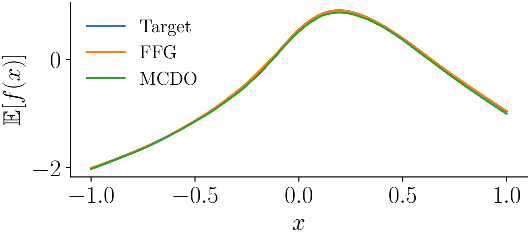


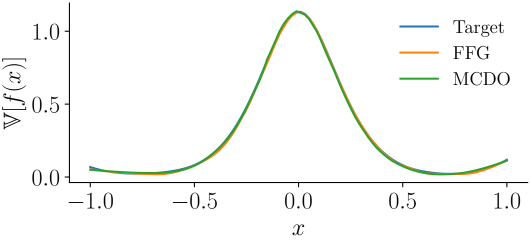
with $b = \min_{x \in A} m(x)$.

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

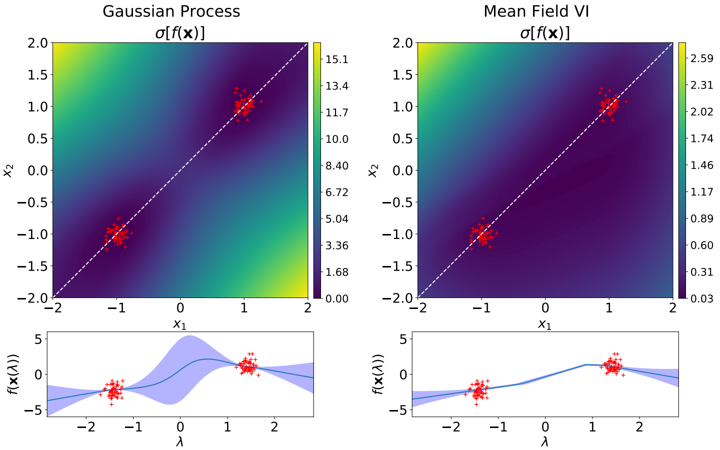
Try to fit mean and variance function from before, but with 2HL net:





Variational Inference in Deep Nets

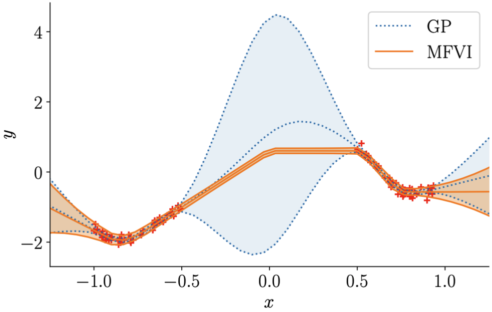
Does theorem 3 imply good uncertainty quantification with VI in deep BNNs?



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!