'In-Between' Uncertainty in Bayesian Neural Networks

Andrew Foong[1], Yingzhen Li[2], José Miguel Hernández-Lobato[1 2 3] and Richard E. Turner[1 2]
[1]University of Cambridge, [2]Microsoft Research Cambridge & [3]Alan Turing Institute

14th June 2019

UNIVERSITY OF CAMBRIDGE

Why we need good uncertainties

Neural networks extremely successful. But overconfident.

Hampers performance in:

- Reinforcement learning
- High-risk decision-making

Neural networks should know what they don't know.

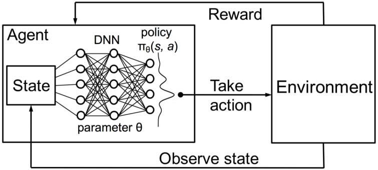


Figure 1: Reinforcement learning: Mao et al. [2016]

Figure 2: Medical diagnosis

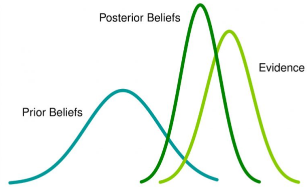
Bayesian neural networks

Standard neural networks learn point estimate of weights.
Bayesian neural networks learn posterior distribution of weights.

Uncertainty in parameters $\rightarrow$ Uncertainty in predictions

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta),$$

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \int p(y^*|\mathbf{x}^*, \theta)p(\theta|\mathcal{D})\,\mathrm{d}\theta.$$



Intractable - *can we approximate it scalably*?

Mean-Field Variational Inference (MFVI)

Variational inference: $p(\theta|\mathcal{D}) \approx q(\theta)$.
Mean field: $q(\theta)$ is factorised Gaussian.

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

Find 'best' $q(\theta)$ by minimising $\mathrm{KL}(q(\theta)||p(\theta|\mathcal{D}))$.

By maximising Evidence Lower Bound (ELBO):

$$\mathrm{ELBO} = \sum_{n=1}^{N} \mathbb{E}_q[\log p(y_n|\mathbf{x}_n, \theta)] - \mathrm{KL}(q(\theta)||p(\theta))$$

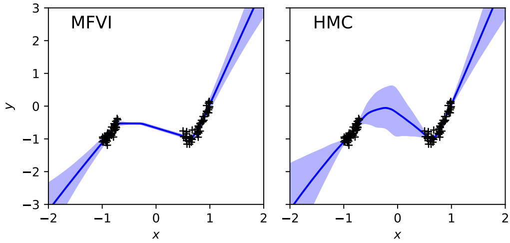
Get gradient of ELBO via Monte Carlo and the reparametrisation trick.

Does MFVI work?

MFVI gives state-of-the-art log-likelihoods on UCI regression. - Tomczak et al. [2018]
But does poorly on contextual bandits, which requires good uncertainties. - Riquelme et al. [2018]

What's going on?

Simple sanity check - 1D regression:



MFVI has uncertainty outside, but not in-between clusters of data.

Lack of ‘in-between’ uncertainty

Two reasons why:

1. Need dependencies to have in-between uncertainty *and* fit data. Simple one-neuron network with fixed output weights, $y = \mathrm{ReLU}(Wx + b)$:

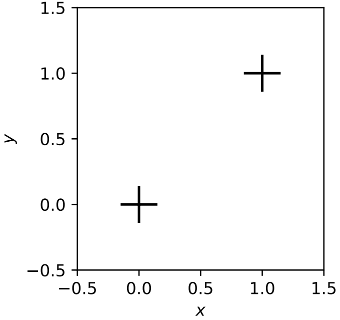


Figure 3: Varying ‘kink’ position while fitting data requires **coordination** between bias and weight.

Lack of 'in-between' uncertainty

Two reasons why:

1. Need dependencies to have in-between uncertainty *and* fit data. Simple one-neuron network with fixed output weights, $y = \mathrm{ReLU}(Wx + b)$:

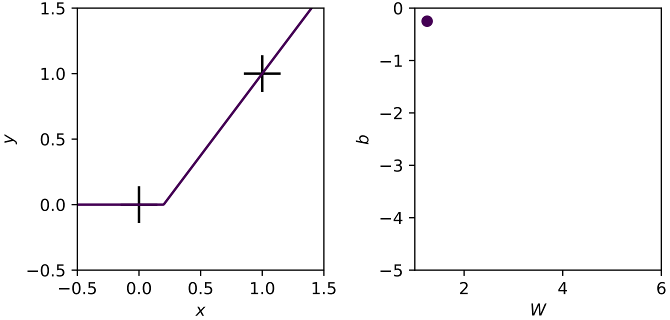


Figure 3: Varying 'kink' position while fitting data requires **coordination** between bias and weight.

Lack of ‘in-between’ uncertainty

Two reasons why:

1. Need dependencies to have in-between uncertainty *and* fit data.
   Simple one-neuron network with fixed output weights,
   $y = \mathrm{ReLU}(Wx + b)$:

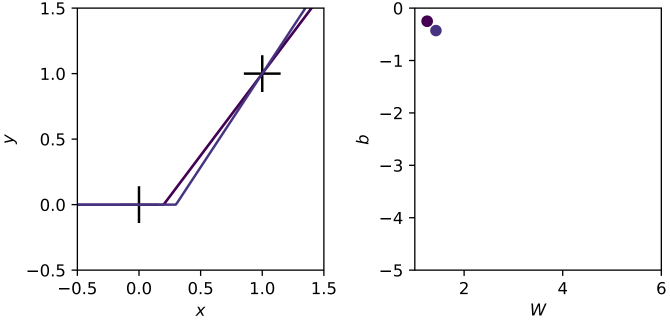


Figure 3: Varying ‘kink’ position while fitting data requires **coordination** between bias and weight.

Lack of 'in-between' uncertainty

Two reasons why:

1. Need dependencies to have in-between uncertainty *and* fit data. Simple one-neuron network with fixed output weights, $y = \mathrm{ReLU}(Wx + b)$:

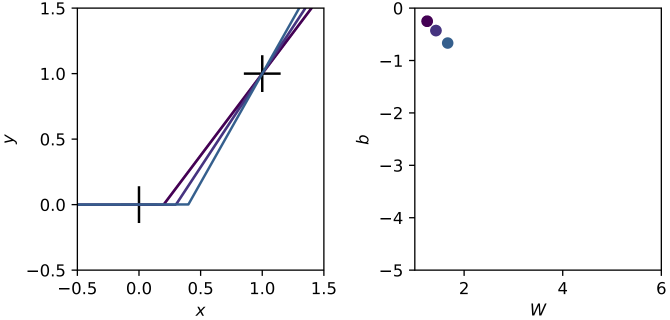


Figure 3: Varying 'kink' position while fitting data requires **coordination** between bias and weight.

Lack of 'in-between' uncertainty

Two reasons why:

1. Need dependencies to have in-between uncertainty *and* fit data. Simple one-neuron network with fixed output weights, $y = \mathrm{ReLU}(Wx + b)$:

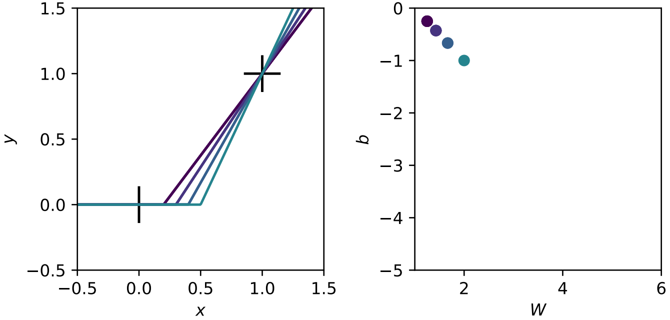


Figure 3: Varying 'kink' position while fitting data requires **coordination** between bias and weight.

Lack of 'in-between' uncertainty

Two reasons why:

1. Need dependencies to have in-between uncertainty *and* fit data. Simple one-neuron network with fixed output weights, $y = \mathrm{ReLU}(Wx + b)$:

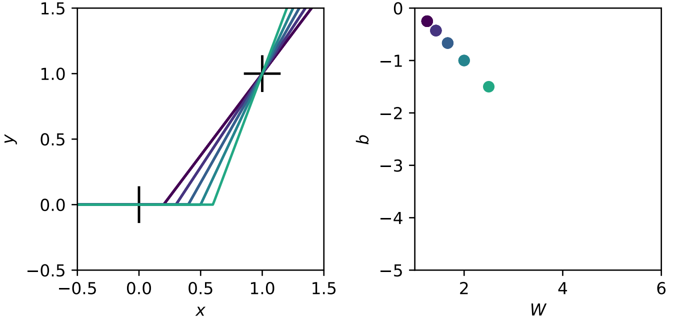


Figure 3: Varying 'kink' position while fitting data requires **coordination** between bias and weight.

Lack of 'in-between' uncertainty

Two reasons why:

1. Need dependencies to have in-between uncertainty *and* fit data. Simple one-neuron network with fixed output weights, $y = \mathrm{ReLU}(Wx + b)$:

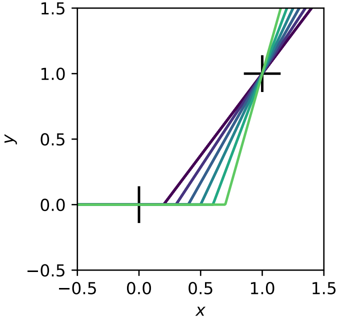


Figure 3: Varying 'kink' position while fitting data requires **coordination** between bias and weight.

Lack of 'in-between' uncertainty

Two reasons why:

1. Need dependencies to have in-between uncertainty *and* fit data. Simple one-neuron network with fixed output weights, $y = \mathrm{ReLU}(Wx + b)$:

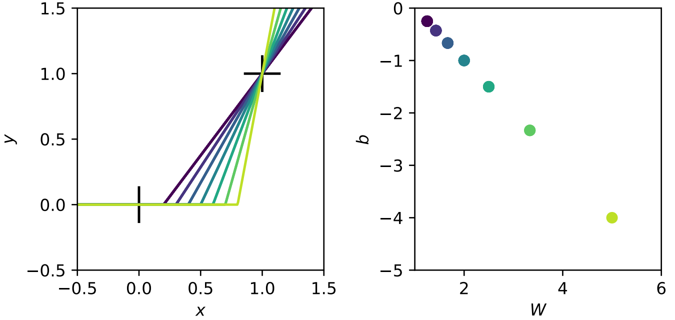


Figure 3: Varying 'kink' position while fitting data requires **coordination** between bias and weight.

Lack of ‘in-between’ uncertainty

Two reasons why:

2. We show that mean-field causes *convex variance* in a simplified case - $\mathrm{Var}[f_\theta(\mathbf{x})]$ is convex in $\mathbf{x}$!

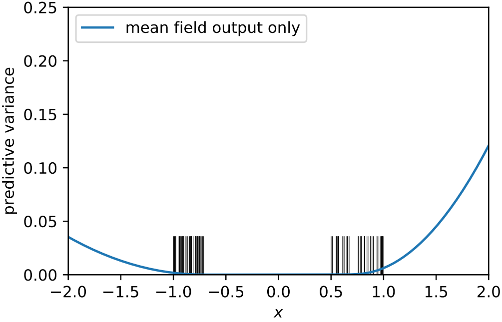


Figure 4: In a single hidden layer ReLU NN, **mean-field leads to convex uncertainty** when being Bayesian over only output weights.

Lack of 'in-between' uncertainty

Two reasons why:

2. We show that mean-field causes *convex variance* in a simplified case - $\mathrm{Var}[f_\theta(\mathbf{x})]$ is convex in $\mathbf{x}$!

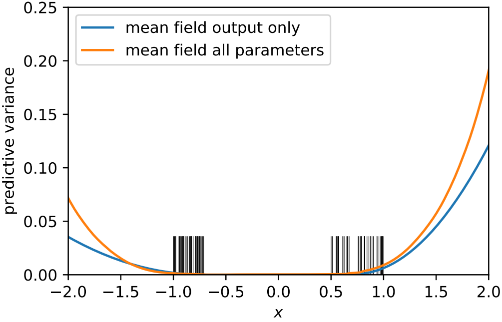


Figure 4: In a single hidden layer ReLU NN, **mean-field leads to convex uncertainty** when being Bayesian over only output weights.

Lack of 'in-between' uncertainty

Two reasons why:

2. We show that mean-field causes *convex variance* in a simplified case - $\mathrm{Var}[f_\theta(\mathbf{x})]$ is convex in $\mathbf{x}$!

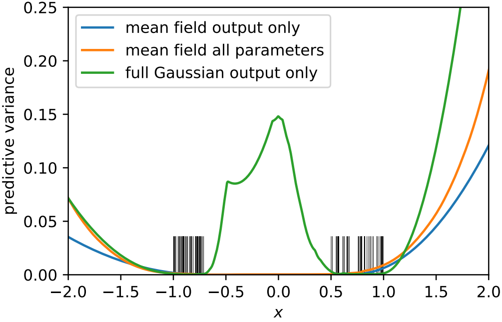
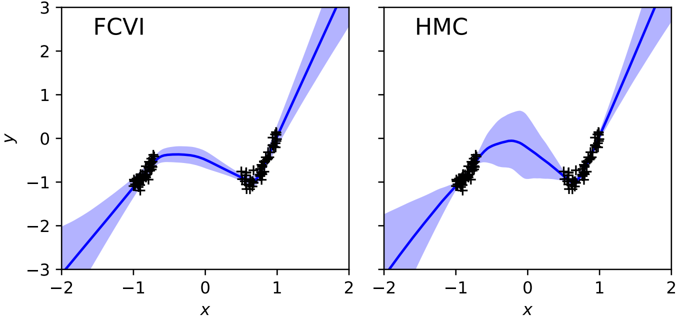


Figure 4: In a single hidden layer ReLU NN, **mean-field leads to convex uncertainty** when being Bayesian over only output weights.

What about full covariance VI (FCVI)?

Could also optimise entire covariance matrix using VI. Better, but:

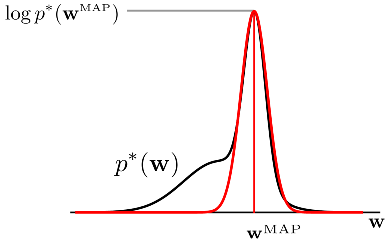
- difficult to optimise.
- still overconfident in-between.



Back to a classical full-covariance technique

Laplace approximation - one of the first BNN methods - MacKay [1992].

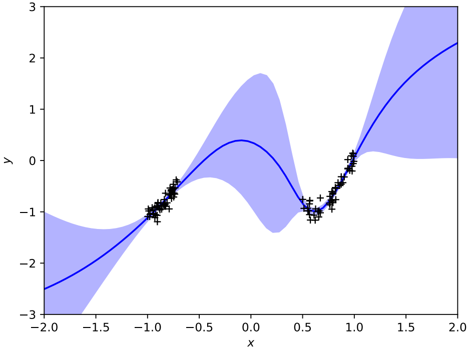
- Find mode of posterior.
- Estimate curvature there and fit a full-covariance Gaussian.
- Linearise output of the network.
- Solve linear Gaussian model to make predictions.



Can this classical technique provide in-between uncertainty?

Laplace approximation - 1D performance

Yes it can! (Have to use tanh activations for nice linearisation).



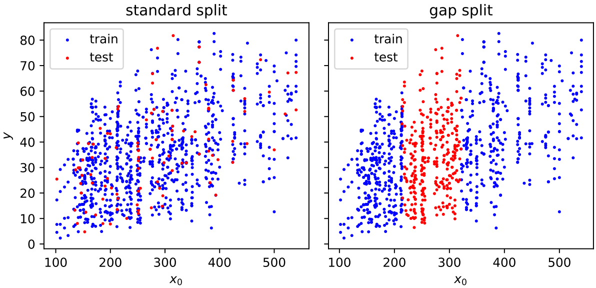
Does this observation extend to higher dimensional datasets?

UCI datasets revisited

UCI a popular BNN benchmark: Hernández-Lobato and Adams [2015]

Standard splits uniformly sample test set - doesn't test in-between uncertainty.

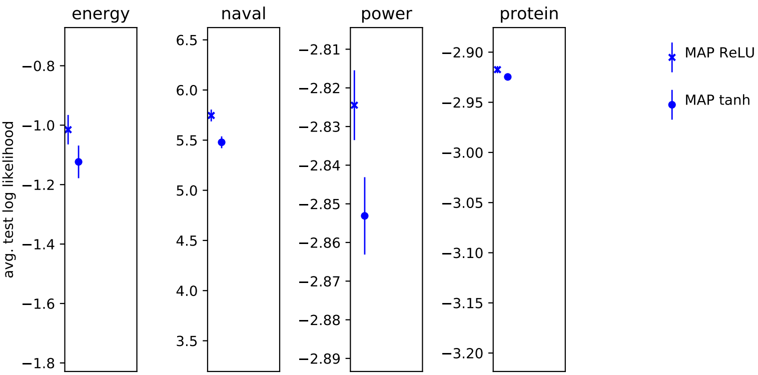
We create new splits with middle third as test set.



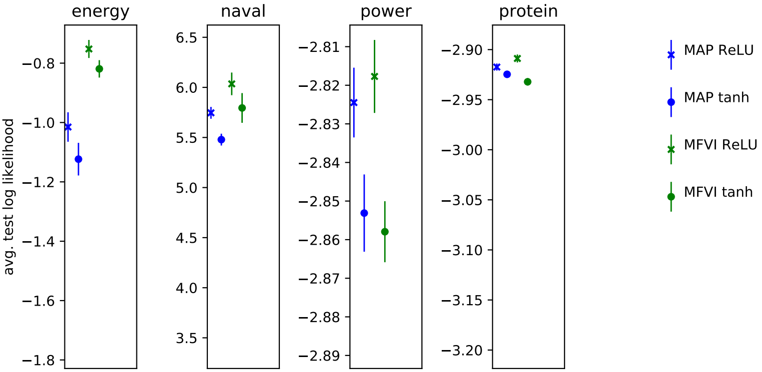
A good method must:

- Do well on standard splits - fit the data.
- Not fail catastrophically on gap splits - not overconfident.

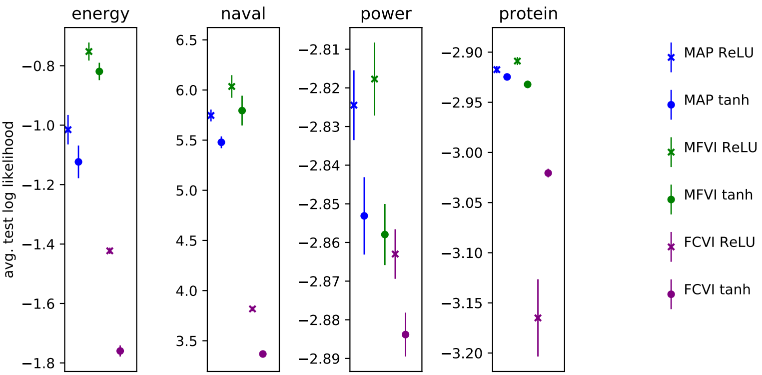
Standard split UCI results - higher is better



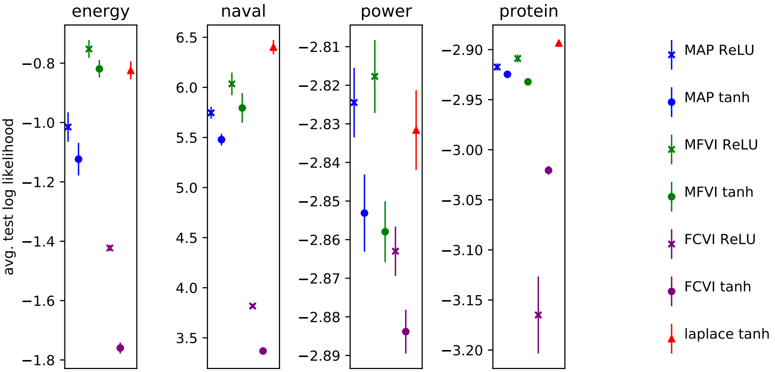
Standard split UCI results - higher is better



Standard split UCI results - higher is better

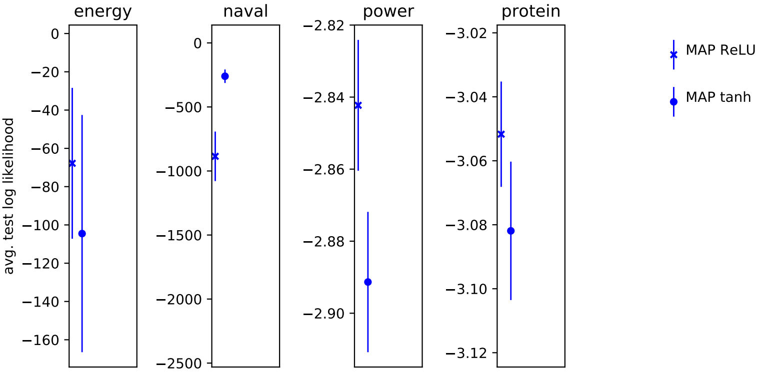


Standard split UCI results - higher is better



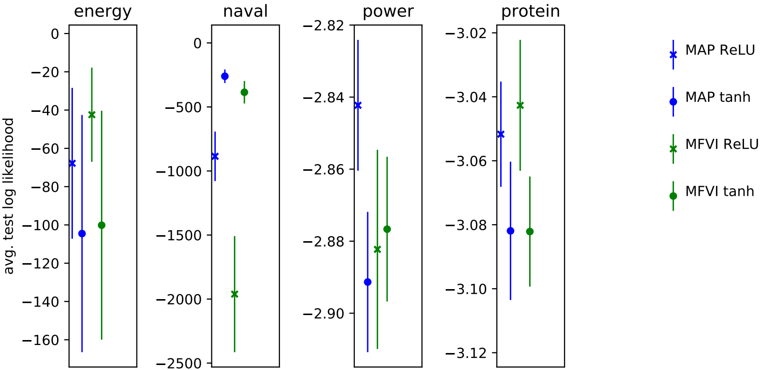
No clear winner, but FCVI does poorly.

Gap split UCI results - higher is better



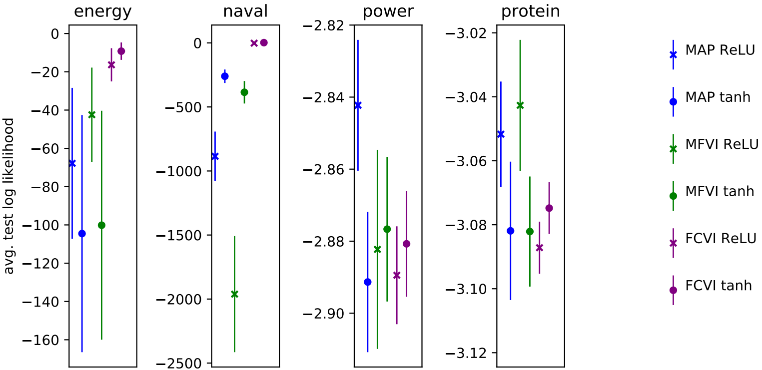
MAP fails catastrophically on energy and naval.

Gap split UCI results - higher is better



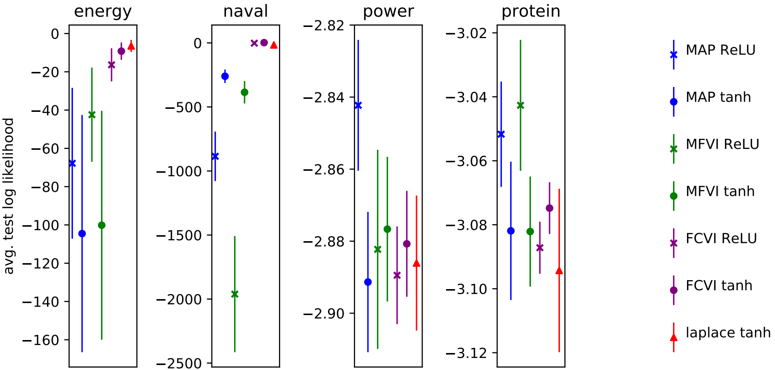
MAP fails catastrophically on energy and naval. So does MFVI!

Gap split UCI results - higher is better



MAP fails catastrophically on energy and naval. So does MFVI!
FCVI does better - unsurprising as it underfits.

Gap split UCI results - higher is better



MAP fails catastrophically on energy and naval. So does MFVI!
FCVI does better - unsurprising as it underfits.
Only Laplace does well on standard and gap splits.

Conclusions

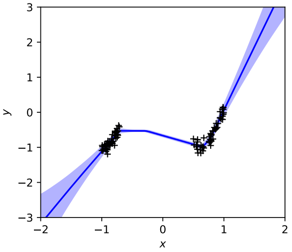
MFVI fails to provide in-between uncertainty.

Standard UCI fails to test for it.

Less scalable classical methods do provide it.

Take home message: Think about how approximations in parameter space restrict the expressiveness of uncertainty in function space.

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



Acknowledgements

We thank David R. Burt, Sebastian W. Ober and Ross Clarke for helpful discussions.

And I'd like to thank the **Trinity Hall Research Studentship** and the **George and Lilian Schiff Foundation** for funding my studies.

Thanks for listening!

Image credits: Figure 2 taken from `https://www.iqvis.com/blog/`.

References

J. M. Hernández-Lobato and R. Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

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

H. Mao, M. Alizadeh, I. Menache, and S. Kandula. Resource management with deep reinforcement learning. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, pages 50–56. ACM, 2016.

C. Riquelme, G. Tucker, and J. Snoek. Deep Bayesian bandits showdown: An empirical comparison of Bayesian deep networks for Thompson sampling. *arXiv preprint arXiv:1802.09127*, 2018.

M. B. Tomczak, S. Swaroop, and R. E. Turner. Neural network ensembles and variational inference revisited. In *1st Symposium on Advances in Approximate Bayesian Inference*, pages 1–11, 2018.