Linearised neural networks

CBL reading group: Bayesian Neural Networks, 22 March 2023

James Allingham, Javier Antoran, Vincent Fortuin

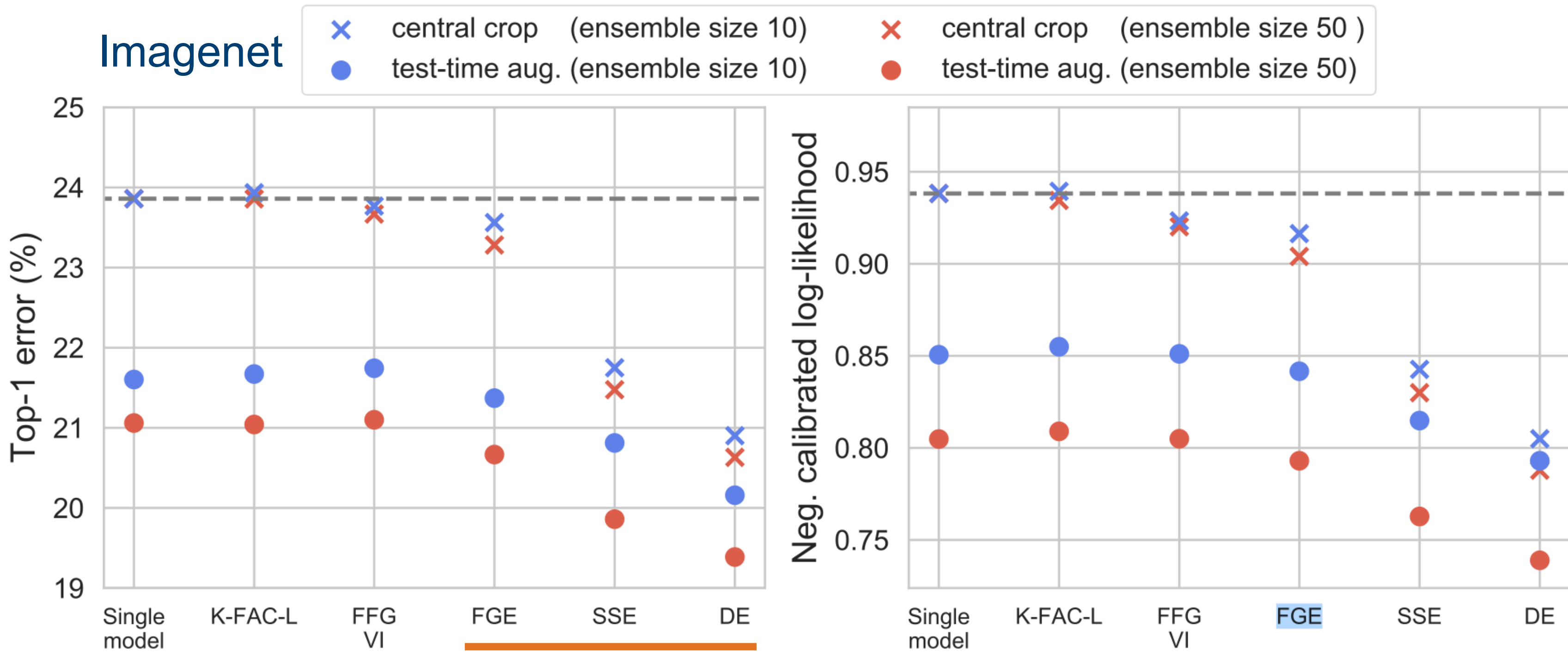
UNIVERSITY OF CAMBRIDGE

Section structure

1. Motivation: the state of Bayesian deep learning
2. Mackay’s Linearised Laplace method
3. A modern take on Linearised Laplace
4. Linearised Laplace approximation for model selection
5. Scalability of Linearised Laplace
6. Infinitely wide neural networks

Motivation: Bayesian Deep Learning around 2019

- Ensembles works for uncertainty estimation, everything else doesn't



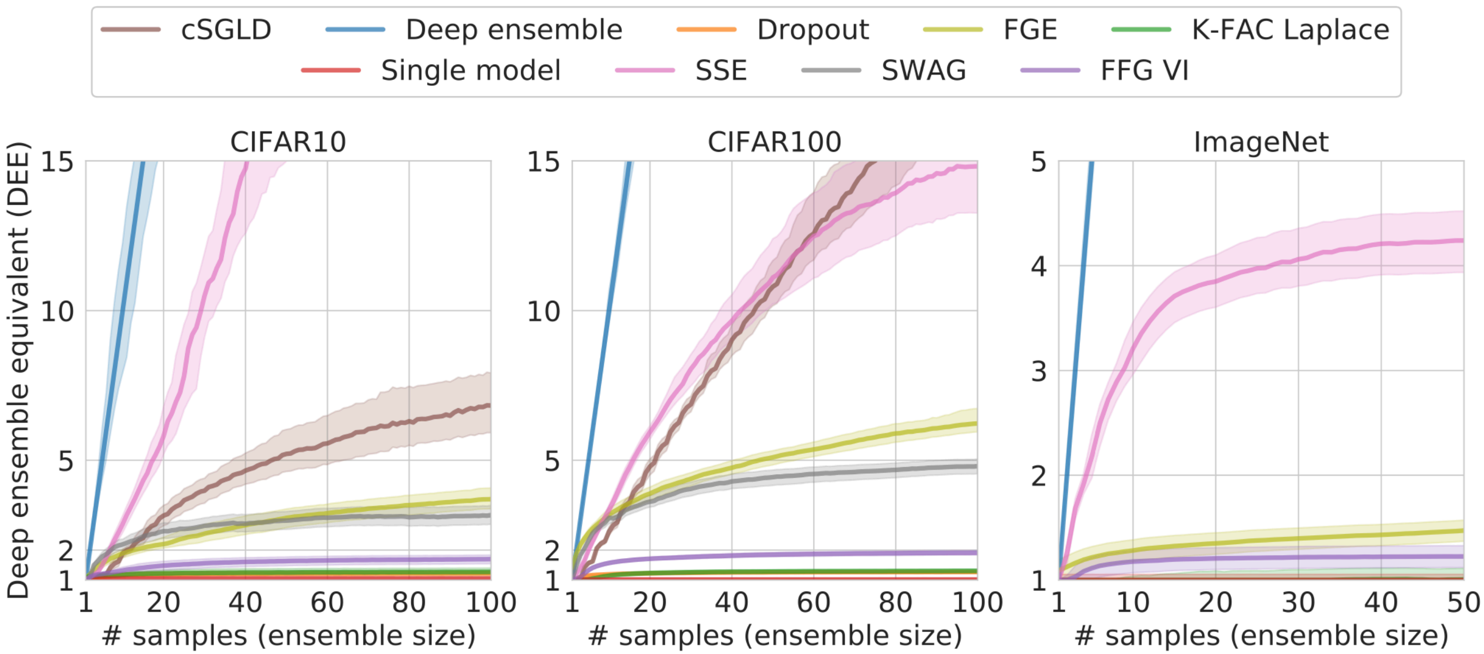
Methods that improve over single model are ensembles

Ashukha et. al. ICLR 2020

Motivation: Bayesian Deep Learning around 2019

- Ensembles works for uncertainty estimation, everything else doesn't

Ensemble "equivalent" score



Ashukha et. al. ICLR 2020

Ensembles give poor joint predictions

ResNet18 + CIFAR100

Line n

	κ	MAP	Ensemble (5)
marginal LL	1	-1.40 $\pm$ 0.00	**-0.90 $\pm$ 0.00**
joint LL	2	-13.97 $\pm$ 0.01	-6.86 $\pm$ 0.01
	3	-27.89 $\pm$ 0.03	-14.17 $\pm$ 0.03
	4	-41.83 $\pm$ 0.03	-22.29 $\pm$ 0.04
	5	-55.89 $\pm$ 0.02	-31.07 $\pm$ 0.09

Osband et. al. 2022. Antorán, Padhy, et. al. 2023

Notation slide

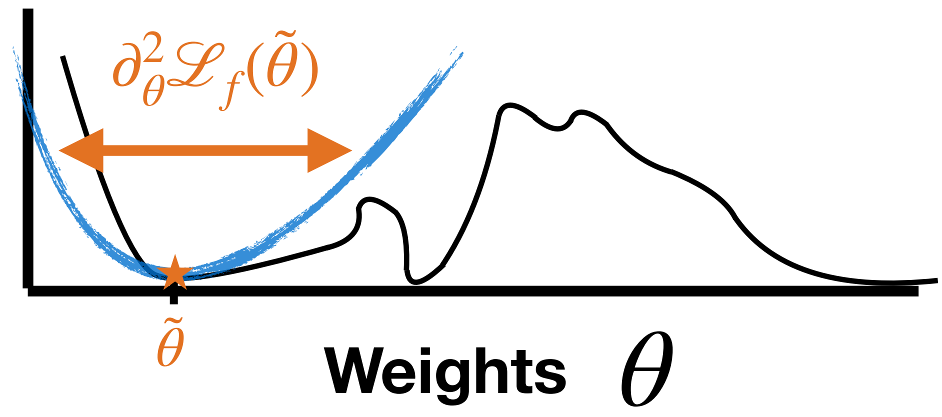
- Select a NN function $f : \Theta \times \mathscr{X} \rightarrow \mathscr{Y}$ and place a prior distribution $\pi(\theta)$ over NN parameters.
- Define some likelihood function $p(y\,|\,f(\theta, x))$ to characterise the agreement of the NN function with the observations (y, x)
- Posterior over parameters is given by

$$p(\theta\,|\,x, y) = \frac{exp(-\mathscr{L}_f(\theta))}{Z}$$

- Where $\mathscr{L}_f(\theta) = -\underbrace{\log p(y\,|\,f(\theta, x))}_{\text{data fit}} - \underbrace{\log \pi(\theta)}_{\text{regulariser}}$

Linearised Laplace (Bayesian methods for adaptive models, 1991)

$$\mathscr{L}_f(\theta) = \underbrace{-\log p(y\,|\,f(\theta, x))}_{\text{data fit}} \underbrace{-\log \pi(\theta)}_{\text{regulariser}}$$



$$\tilde{\theta} \in \text{argmin}_\theta\, \mathscr{L}_f(\theta)$$

$$p(\theta\,|\,x, y) \approx \mathcal{N}(\tilde{\theta}, (\partial_\theta^2 \mathscr{L}_f(\tilde{\theta}))^{-1})$$

Linearisation as an approximation to the predictive

Predictive distribution intractable:

$$\int f(\theta, x^*)\ \mathcal{N}(\theta; \tilde{\theta}, (\partial_\theta^2 \mathcal{L}_f(\tilde{\theta}))^{-1})\, d\theta \quad \approx$$

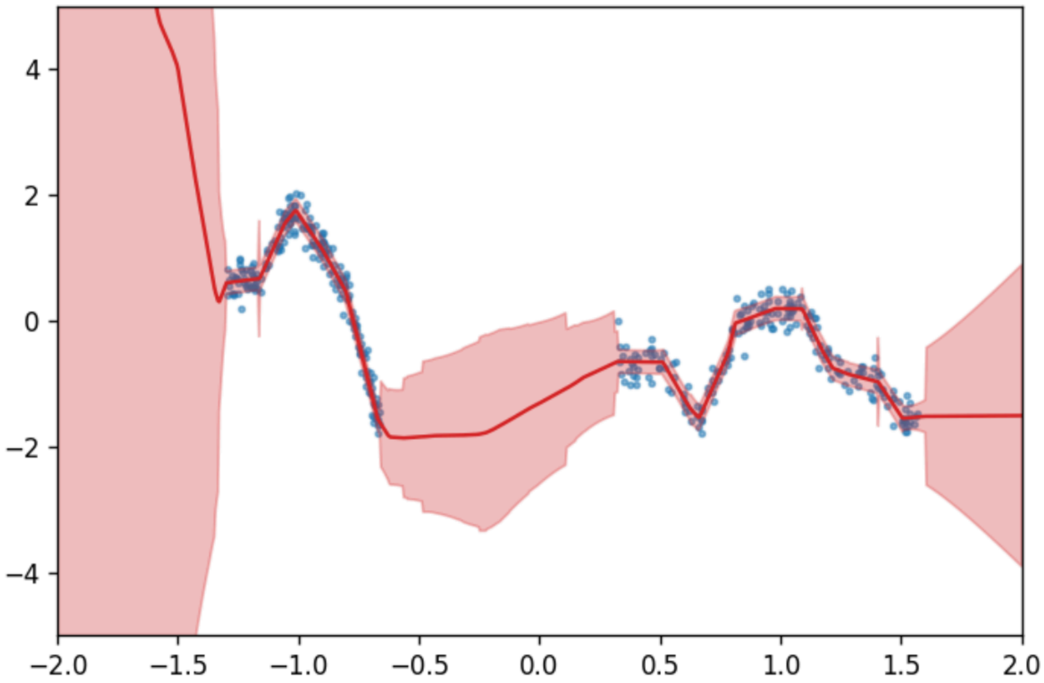
Idea: linearise f $\qquad f(\theta, x) \approx f(\tilde{\theta}, x) + J(x)(\theta - \tilde{\theta})$

$$J(x) = \partial_\theta f(\tilde{\theta}, x)$$

$$\approx \mathcal{N}(f(\tilde{\theta}, x^*),\ J(\partial_\theta^2 \mathcal{L}_f(\tilde{\theta}))^{-1} J^T)$$

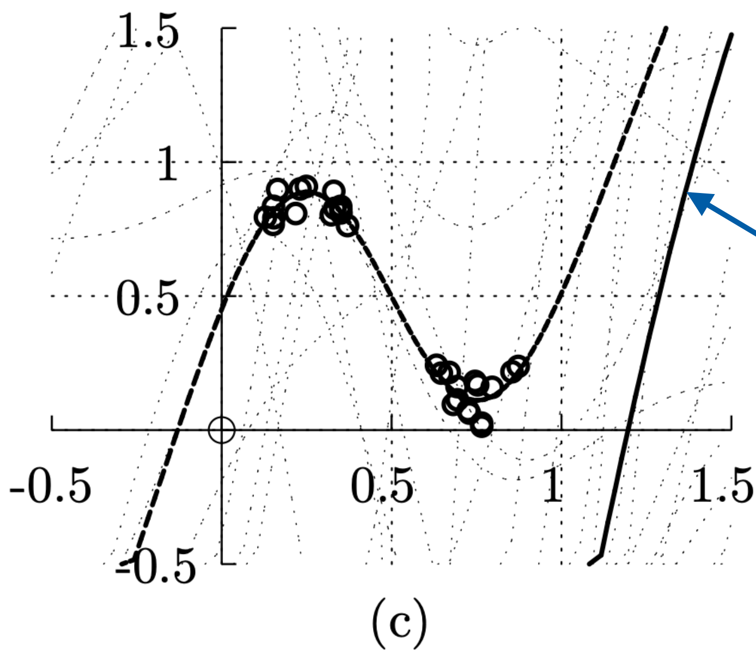
Linearised Laplace uncertainty: examples

$$\approx \mathcal{N}(f(\tilde{\theta}, x^*), J(\partial_\theta^2 \mathcal{L}_f(\tilde{\theta}))^{-1} J^T)$$



What if we don't linearise?

$$\int f(\theta, x^*)\ \mathcal{N}(\theta; \tilde{\theta}, (\partial^2_\theta \mathcal{L}_f(\tilde{\theta}))^{-1})\, d\theta$$

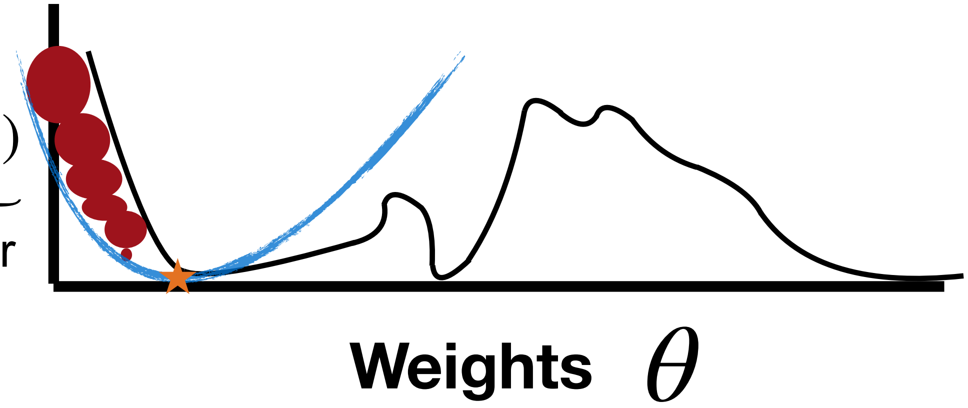


Predictive mean

Neil D. Lawrence, PhD Thesis

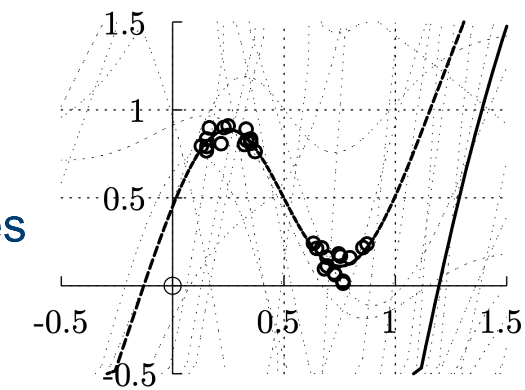
What went wrong?

$$\mathscr{L}_f(\theta) = \underbrace{-\log p(y\,|\,f(\theta, x))}_{\text{data fit}} \underbrace{-\log \pi(\theta)}_{\text{regulariser}}$$



Issue: A lot of mass falls in low density region, leading to bad predictions

Leads to crazy posterior samples



(c)

Solution: view the linearisation as a model change

If we linearise f $\quad f(\theta, x) \approx f(\tilde{\theta}, x) + J(x)(\theta - \tilde{\theta}) \doteq h(\theta, x)$

We may consider $\quad y = h(\theta, x) + \epsilon \quad$ and $\quad \begin{matrix} \theta \sim \mathcal{N}(0, A^{-1}) \\ \epsilon \sim \mathcal{N}(0, B^{-1}) \end{matrix}$

With true linear model posterior $\theta \,|\, x, y \sim \mathcal{N}(\tilde{\theta}, H^{-1})$

$$H \doteq J^T B J + A \quad \approx \partial^2_\theta \mathcal{L}_f(\tilde{\theta})$$

Known as the Generalised
Gauss Newton approximation

This linear model has the NN mean and linear-Gaussian error-bars

$$\mathcal{N}(f(\tilde{\theta}, x),\; J H^{-1} J^T)$$

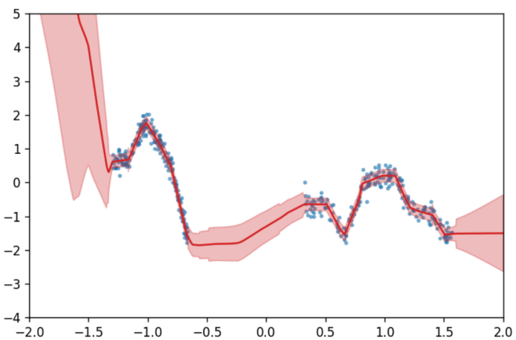
Khan et. al. 2019, Immer et. al. 2021

Remaining issue: choosing a regulariser

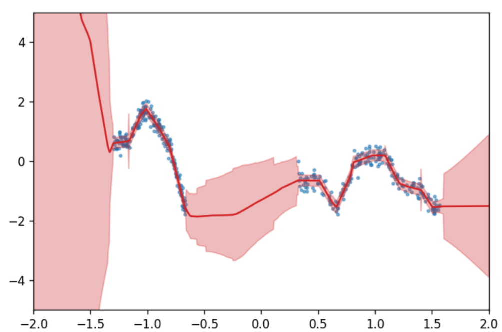
$$A = \lambda I$$

2 hidden layer, 2600 parameter, MLP with batchnorm

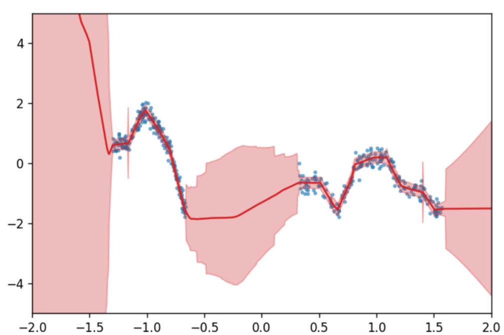
$\lambda = 100$



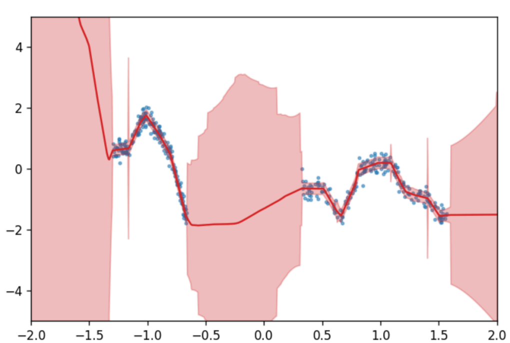
$\lambda = 10$



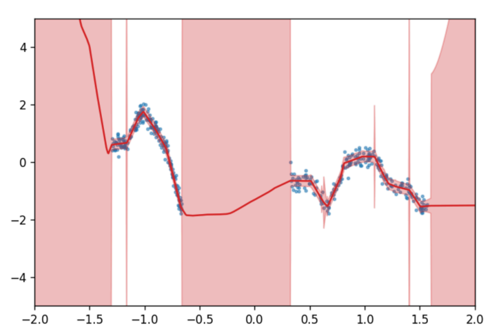
$\lambda = 5$



$\lambda = 1$



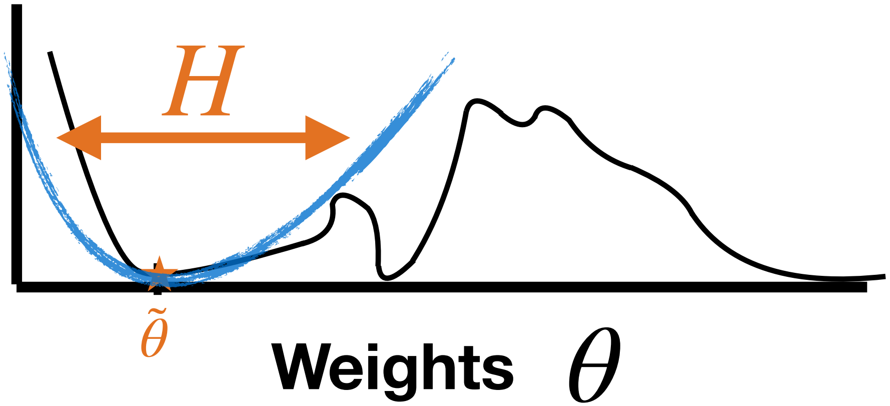
$\lambda = 0.1$



Mackay's solution: Iterative algorithm

1. Minimise NN Loss

$$\mathcal{L}_f(\theta, A1) = \underbrace{-\log p(y \mid f(\theta, x))}_{\text{data fit}} + \underbrace{\|\theta\|_{A1}}_{\text{regulariser}}$$



2. Choose regulariser to maximise posterior volume

$$A_2 = \text{argmax}_A \underbrace{-\mathcal{L}(\tilde{\theta}, A) - \log det(H) + C}_{\doteq \mathcal{M}(A)}$$

3. Retrain NN: i.e. goto 1.

Immer et. al. 2021's online approach

1. Optimise NN loss for s few steps

$$\mathscr{L}_f(\theta) = -\log p(y \,|\, f(\theta, x)) + \|\theta\|_A$$

2. Single step of evidence update
at current weights

$$-\mathscr{L}(\theta, A) - \text{l}ogdet(H(\theta)) + C$$

3. Retrain NN: i.e. goto 1.

Interpretation of quadratic expansion around an optima of the loss is lost

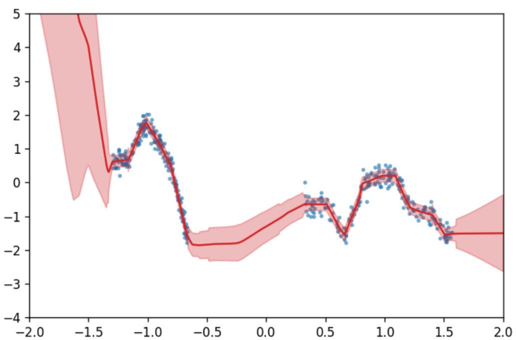
Same procedure, with different derivation was also used by Friston et. al. 2006 for neuroimaging. They called it 'Variational Laplace'.

Some pathologies arise; post-hoc setting

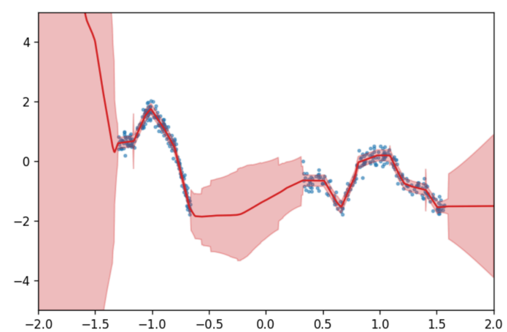
$$\Lambda = \lambda I$$

2 hidden layer, 2600 parameter, MLP with batchnorm

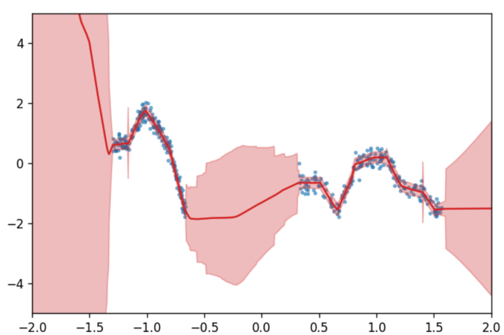
$\lambda = 100$



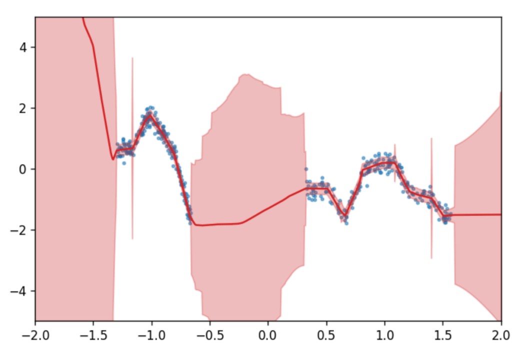
$\lambda = 10$



$\lambda = 5$

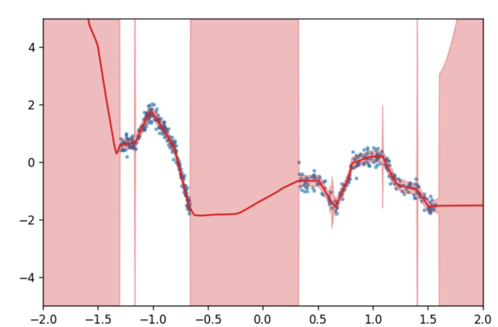


$\lambda = 1$



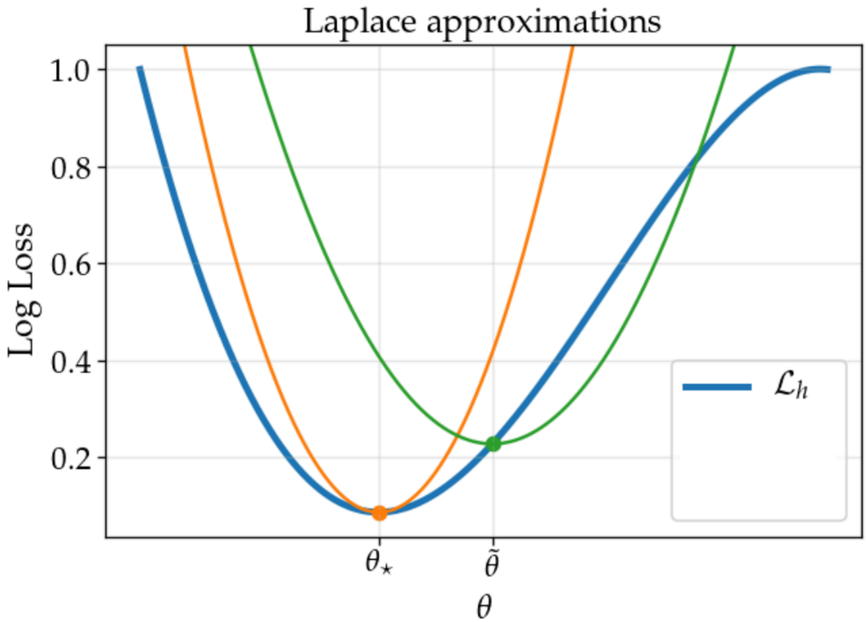
Largest $\mathscr{M}$

$\lambda = 0.1$

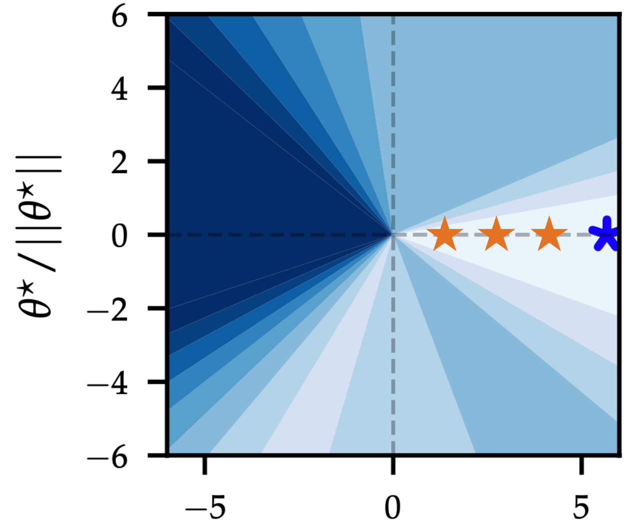


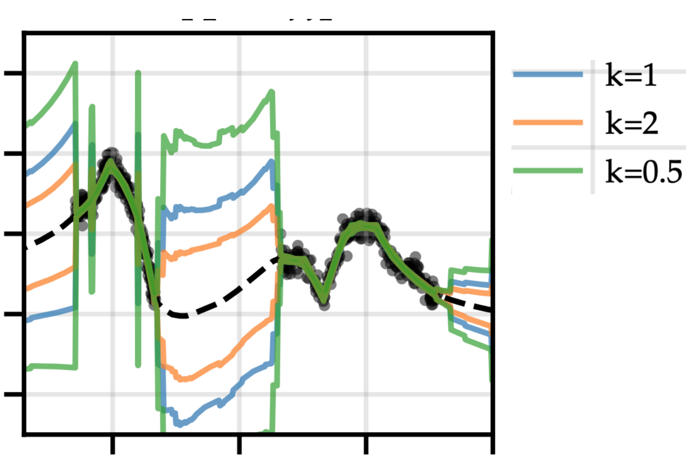
What is wrong with the Laplace model evidence?

$$\tilde{\theta} \not\in \text{argmin}_\theta \mathscr{L}_f(\theta)$$



$$\log p(y \,|\, f(\theta, x))$$



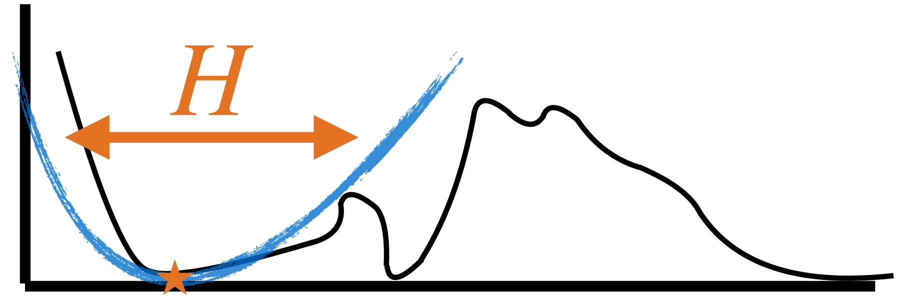


$$\mathscr{L}_f(\theta) = \underbrace{-\log p(y \,|\, f(\theta, x))}_{\text{invariant}} + \underbrace{||\theta||^2_\Lambda}_{\text{not invariant}}$$

Antorán et. al. 2022

Limitation: scalability

$$H \in \mathcal{R}^{|\Theta| \times |\Theta|}$$



Is intractable to store when $|\Theta|$ is large

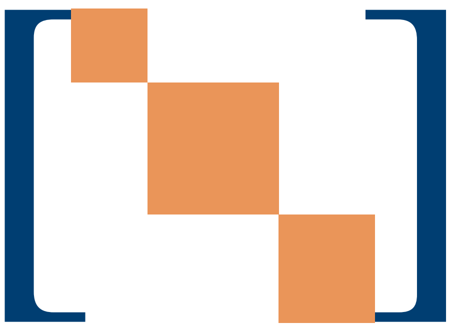
Predictive distribution $\mathcal{N}(f(\tilde{\theta}, x),\ JH^{-1}J^T)$

Evidence $-\mathcal{L}(\theta, A) - \log det(H(\theta)) + C$

Both $\mathcal{O}(|\Theta|^3)$

Different approaches to scalability

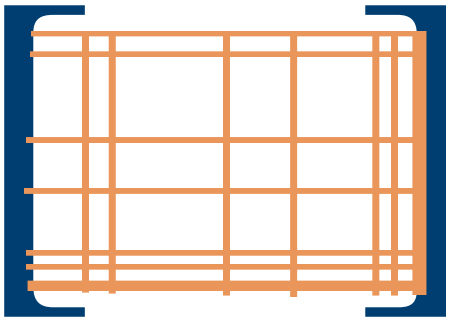
KFAC



$$\sum_{(x,y)} ab \approx \left(\sum a\right)\left(\sum b\right)$$

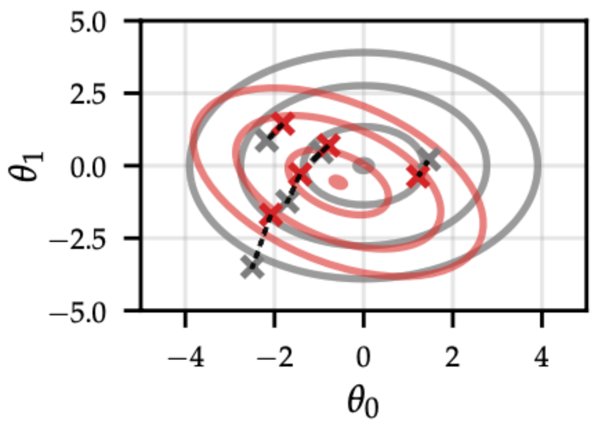
- Overestimates curvature

Subnetwork



- Selecting weights is hard
- Requires manual tuning

Sampling



- Unbiased & performance close to full-covariance Laplace
- Best regulariser selection

Last-Layer / Last-Layer + KFAC

$$f^{L-1}(\theta, x) = J^{L-1}(x)$$

- Avoids dealing with full $J(x)$
- Best uncertainty
- Fastest
- Requires manual tuning

Infinite width NNs

As NN width goes to infinity, assuming $\theta \sim \mathcal{N}(0, A^{-1})$ for properly scaled A

$$f(\theta, \cdot\,) \sim GP(0, K(\,\cdot\,,\cdot\,))$$

$$K(\,\cdot\,,\cdot\,) = \mathbb{E}_\theta[f(\theta, \cdot\,)f(\theta, \cdot\,)] = J^{L-1}(\,\cdot\,)(J^{L-1}(\,\cdot\,))^T$$

Kernel is outer product of last layer Jacobians

de G. Matthews et. al, 2018 & Lee et. al. 2018

Convergence to limiting behaviour can be fast

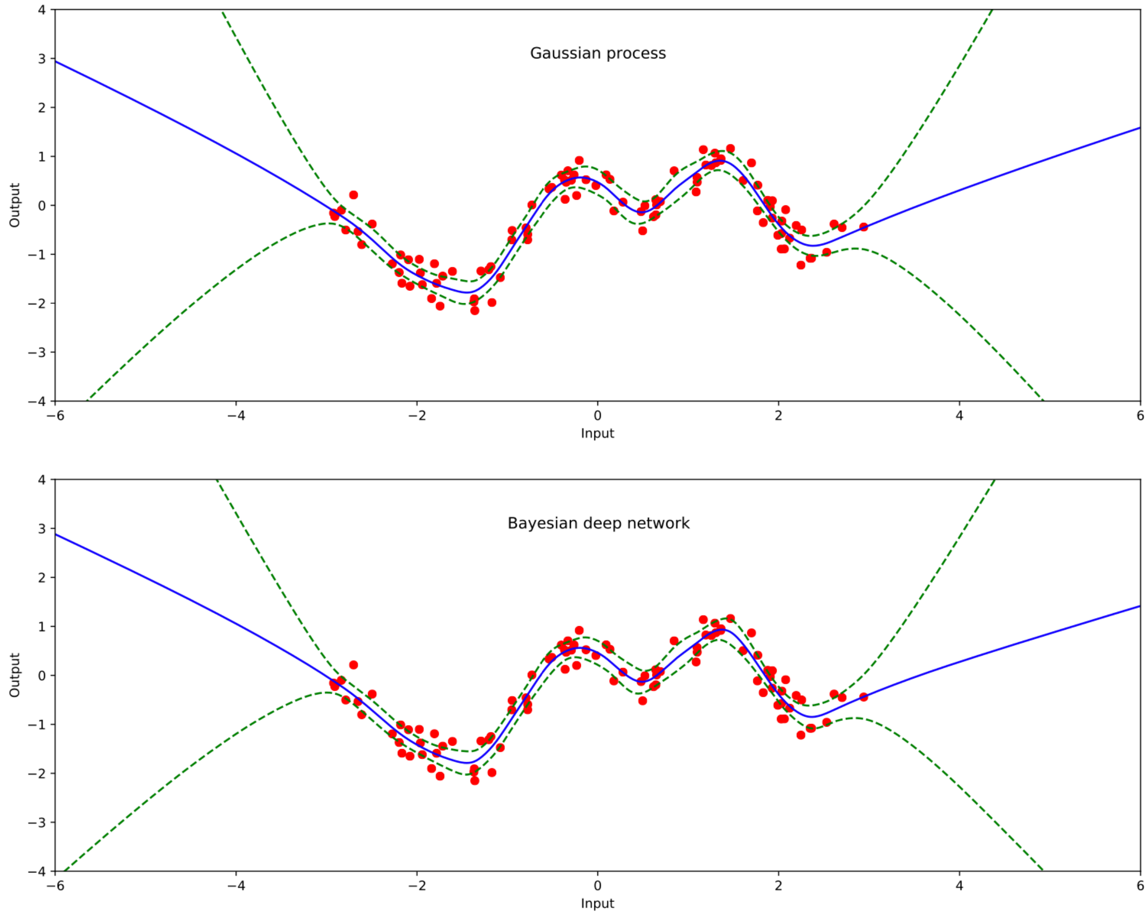


Figure 6: A comparison between Bayesian posterior inference in a Bayesian deep neural network and posterior inference in the analogous Gaussian process for the Snelson dataset. The neural network has 3 hidden layers and 50 units per layer. The lines show the posterior mean and two σ credible intervals.