Bayesian Inference for Deep Learning

Inference and modern trends for Bayesian Neural Networks: Laplace and Ensemble methods

Simone Rossi and Maurizio Filippone

Data Science Department, EURECOM (France)

Approximate inference for Bayesian deep learning

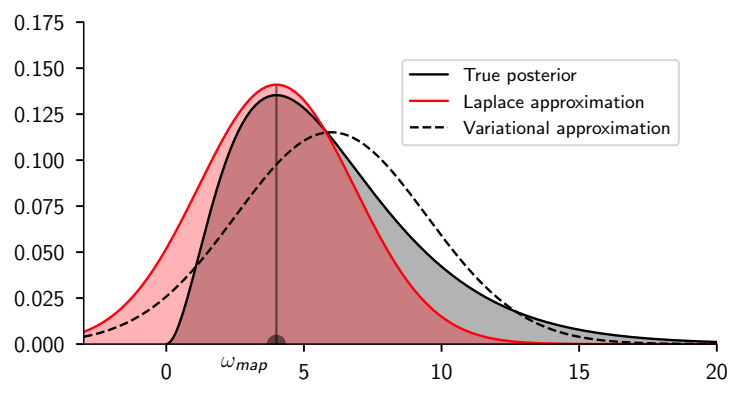
- Collapse the posterior on the most likely value (Maximum-a-Posteriori or MAP)
- Approximate the intractable posterior:
  - Use Variational Inference
  - Use Laplace approximation (local approximation the MAP solution)
- Sample from the intractable posterior:
  - Markov-Chain Monte-Carlo (Hamiltonian Monte-Carlo)

Laplace approximation for Bayesian neural networks

Laplace approximation

Idea: Local approximation around MAP solution

The *Laplace approximation* approximates the posterior by a Gaussian $q(\mathbf{w}) = \mathcal{N}(\mathbf{w}_{map}, \mathbf{\Sigma})$ around the mode $\mathbf{w}_{map}$ with covariance $\mathbf{\Sigma}$



Laplace approximation

- Covariance $\mathbf{\Sigma}$ given by the Hessian of the posterior via Taylor expansion:

$$\mathbf{\Sigma} = -\left[\nabla^2_{\mathbf{ww}} \log p(\mathbf{w} \mid \mathbf{y}, \mathbf{X})\Big|_{\mathbf{w}=\mathbf{w}_{map}}\right]^{-1}$$

- We need to compute the Hessian of

$$\log p(\mathbf{w} \mid \mathbf{y}, \mathbf{X}) \propto \sum_{i=1}^{N} \log p(y_i \mid f(\mathbf{x}_n; \mathbf{w})) + \log p(\mathbf{w}) + \text{const.}$$

  The prior term is easy, the likelihood less.

MacKay (1991). *Bayesian Model Comparison and Backprop Nets.* NeurIPS

Computational tractability of the Hessian (and approximations)

Computed through the *Jacobian* and *Hessian* matrix of the neural network $\mathbf{f} = f(\mathbf{x}; \mathbf{w})$,

$$\nabla^2_{\mathbf{ww}} \log p(y \mid f(\mathbf{x}; \mathbf{w})) = \mathcal{H}_{\mathbf{w}}(\mathbf{x})^\top \nabla_{\mathbf{f}} \log p(\mathbf{y} \mid \mathbf{f}) + \mathcal{J}_{\mathbf{w}}(\mathbf{x})^\top \nabla^2_{\mathbf{ff}} \log p(\mathbf{y} \mid \mathbf{f}) \mathcal{J}_{\mathbf{w}}(\mathbf{x})$$

with $[\mathcal{J}_{\mathbf{w}}(\mathbf{x})]_{ci} = \frac{\partial f_c(\mathbf{x};\mathbf{w})}{\partial \omega_i}$ and $[\mathcal{H}_{\mathbf{w}}(\mathbf{x})]_{cij} = \frac{\partial^2 f_c(\mathbf{x};\mathbf{w})}{\partial \omega_i \partial \omega_j}$.

Schraudolph (2002). *Fast curvature matrix-vector products for second-order gradient descent*. Neural Comput.

Computational tractability of the Hessian (and approximations)

Computed through the *Jacobian* and *Hessian* matrix of the neural network $\mathbf{f} = f(\mathbf{x}; \mathbf{w})$,

$$\nabla^2_{\mathbf{ww}} \log p(y \mid f(\mathbf{x}; \mathbf{w})) = \mathcal{H}_{\mathbf{w}}(\mathbf{x})^\top \nabla_{\mathbf{f}} \log p(\mathbf{y} \mid \mathbf{f}) + \mathcal{J}_{\mathbf{w}}(\mathbf{x})^\top \nabla^2_{\mathbf{ff}} \log p(\mathbf{y} \mid \mathbf{f}) \mathcal{J}_{\mathbf{w}}(\mathbf{x})$$

with $[\mathcal{J}_{\mathbf{w}}(\mathbf{x})]_{ci} = \frac{\partial f_c(\mathbf{x};\mathbf{w})}{\partial \omega_i}$ and $[\mathcal{H}_{\mathbf{w}}(\mathbf{x})]_{cij} = \frac{\partial^2 f_c(\mathbf{x};\mathbf{w})}{\partial \omega_i \partial \omega_j}$.

Generalized Gauss-Newton approximation

Drop the first term completely

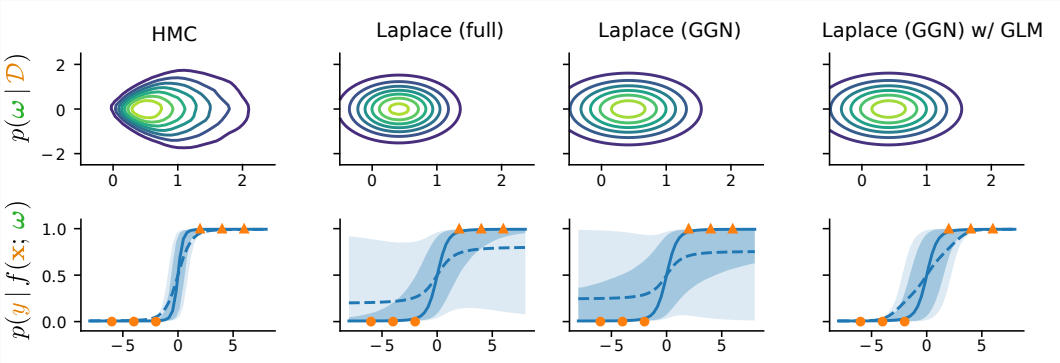
$$\nabla^2_{\mathbf{ww}} \log p(y \mid f(\mathbf{x}; \mathbf{w})) \approx \mathcal{J}_{\mathbf{w}}(\mathbf{x})^\top \nabla^2_{\mathbf{ff}} \log p(\mathbf{y} \mid \mathbf{f}) \mathcal{J}_{\mathbf{w}}(\mathbf{x})$$

This is a good approximation for locally linearized networks.

Schraudolph (2002). *Fast curvature matrix-vector products for second-order gradient descent*. Neural Comput.

Quality of the Laplace approximation

Generally the Laplace approximation can be very rough as it puts a lot of mass in low probability area of the true posterior.



Immer et al. (2021). *Improving predictions of Bayesian neural nets via local linearization*. AISTATS

Laplace approximation for model selection

With Laplace approximation we can also derive another approximation to the marginal likelihood

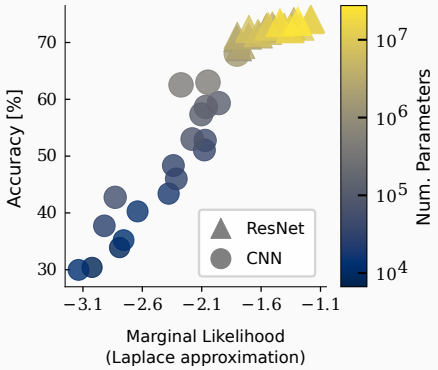
$$\log p(\mathbf{y} \mid \mathbf{X}, \mathcal{M}) \approx p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}_{map}, \mathcal{M}) + \log p(\mathbf{w}_{map} \mid \mathcal{M}) + \frac{1}{2} \log \left| \frac{1}{2\pi} \nabla^2_{\mathbf{ww}} \log p(\mathbf{y}, \mathbf{w} \mid \mathbf{X}, \mathcal{M}) \right|$$

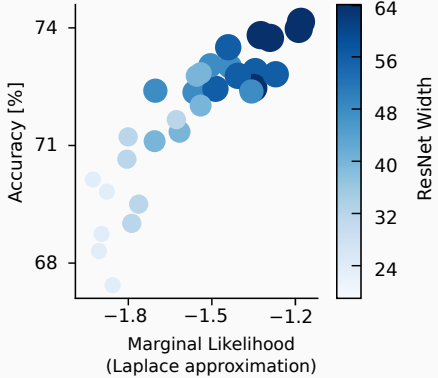
MacKay (1991). *A Practical Bayesian Framework for Backpropagation Networks*. Neural comput.
Immer et al. (2021). *Scalable Marginal Likelihood Estimation for Model Selection in Deep Learning*. ICML

Laplace approximation for model selection

With Laplace approximation we can also derive another approximation to the marginal likelihood

$$\log p(\mathbf{y} \mid \mathbf{X}, \mathcal{M}) \approx p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}_{map}, \mathcal{M}) + \log p(\mathbf{w}_{map} \mid \mathcal{M}) + \frac{1}{2} \log \left| \frac{1}{2\pi} \nabla^2_{\mathbf{ww}} \log p(\mathbf{y}, \mathbf{w} \mid \mathbf{X}, \mathcal{M}) \right|$$





MacKay (1991). *A Practical Bayesian Framework for Backpropagation Networks*. Neural comput.

Immer et al. (2021). *Scalable Marginal Likelihood Estimation for Model Selection in Deep Learning*. ICML

Ensembles methods

A non-Bayesian approach: deep ensembles

Probabilistic, but non-Bayesian, baseline:

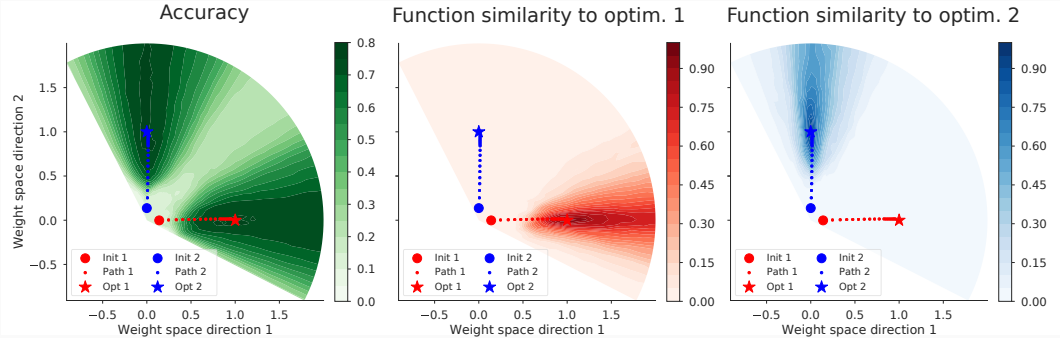
1. Let neural network $f(\mathbf{x}; \mathbf{w})$ parametrize a distribution $p(\mathbf{f})$
   - For regression, the mean $\boldsymbol{\mu}(\mathbf{x}; \mathbf{w})$ and the variance $\boldsymbol{\sigma}(\mathbf{x}; \mathbf{w})$ of a Gaussian distribution
2. Use a proper scoring rule as training criterion
   - For regression, a Gaussian likelihood $p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) = \mathcal{N}(\mathbf{y} \mid \boldsymbol{\mu}(\mathbf{x}; \mathbf{w}), \boldsymbol{\sigma}(\mathbf{x}; \mathbf{w}))$
3. Train an ensemble of M networks with random initialization
4. Combine predictions at test time

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}_i)$$

Note: this is model combination, not Bayesian model average

Lakshminarayanan et al. (2017). *Simple and scalable predictive uncertainty estimation using deep ensembles*. NeurIPS

Deep ensembles explore different modes



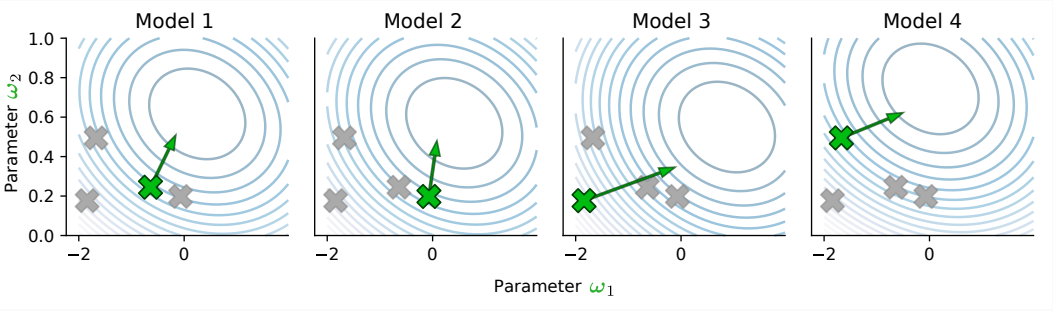
Trajectories of randomly initialized neural networks explore different modes in function space, which explains why deep ensembles trained with just random initializations work well in practice.

Fort et al. (2019). *Deep Ensembles: A Loss Landscape Perspective*. NeurIPS BDL Workshop

A different perspective to ensemble using bootstrap

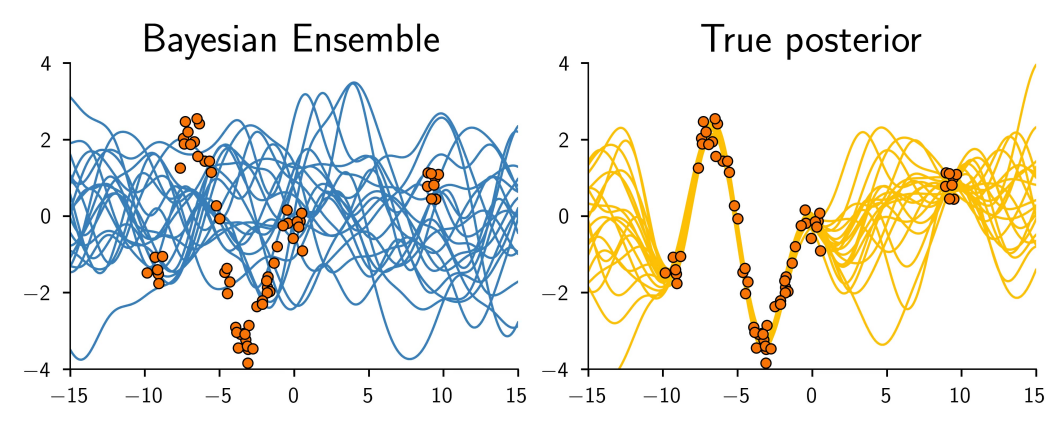
Consider M models with different regularization (e.g. prior) and perturbed data (e.g. likelihood).

1. Perturb the likelihood:
   $p(\mathbf{y} \mid f(\mathbf{x}; \mathbf{w})) = \mathcal{N}(\mathbf{y} \mid f(\mathbf{x}; \mathbf{w}), \sigma^2 I) \xrightarrow{\textit{perturbe}} \mathcal{N}(\tilde{\mathbf{y}}_m \mid f(\mathbf{x}; \mathbf{w}), \sigma^2 I)$ with $\tilde{\mathbf{y}}_m \sim \mathcal{N}(\mathbf{y}, \sigma^2 I)$
2. Perturb the prior:
   $p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{0}, \alpha I) \xrightarrow{\textit{perturbe}} \mathcal{N}(\mathbf{w} \mid \tilde{\mathbf{w}}_m, \alpha I)$ with $\tilde{\mathbf{w}}_m \sim \mathcal{N}(\mathbf{w} \mid \mathbf{0}, \alpha I)$
3. Run MAP on the perturbed model



Bootstrap Ensembles as a Bayesian approximation

Under some conditions, $\{\mathbf{w}_m\}_{m=1}^M$ are samples from the posterior of the original model.



Milios et al. (2020). *Parametric Bootstrap Ensembles as Variational Inference*. AABI

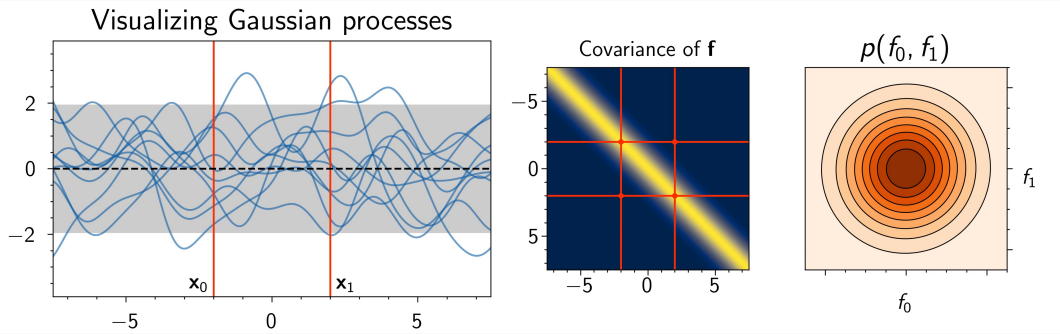
Taking an infinite-limit of neural networks

A quick introduction to Gaussian Processes

*"A **Gaussian process** is a collection of (infinite) random variables, any finite number of which have a joint Gaussian distribution".*

$$f \sim GP(m(\cdot), \kappa(\cdot, \cdot \mid \boldsymbol{\theta})) \quad \Rightarrow \quad p(\mathbf{f}) = \mathcal{N}(\mathbf{m}, \mathbf{K})$$

m is the mean, **K** is the covariance computed for each pair of data with.

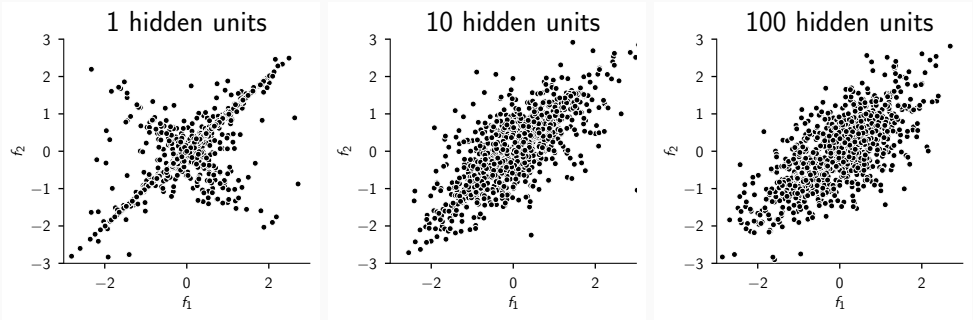


Rasmussen and Williams (2006). *Gaussian Processes for Machine Learning*, MIT Press

Tight connections with shallow neural networks

Bayesian Learning of Neural Networks. R. Neal (1996)

Priors over network parameters can be defined in such a way that the corresponding priors over functions computed by the network reach reasonable limits as the number of hidden units goes to infinity. [...] The infinite network limit also provides insight into the properties of different priors. A Gaussian prior for hidden-to-output weights results in a Gaussian process prior for functions.

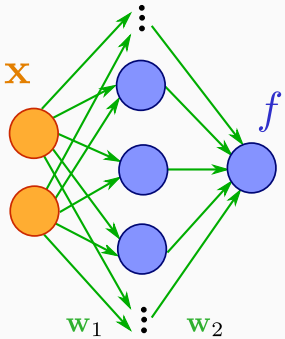


Connection between shallow neural networks and GPs

Gaussian priors on weights: $\mathbf{w}_1 \sim \mathcal{N}(0, \alpha_1 \mathbf{I})$, $\mathbf{w}_2 \sim \mathcal{N}(0, \alpha_2 \mathbf{I})$.
Then $f \sim \mathcal{GP}(0, \mathbf{K})$ in the limit of infinite network width.

$$\begin{aligned}\mathbf{K} = \mathrm{cov}(\mathbf{f}) &= \mathbb{E}_{p(\mathbf{w}_1, \mathbf{w}_2)}[\phi(\mathbf{X}\mathbf{w}_1)\mathbf{w}_2\mathbf{w}_2^\top \phi(\mathbf{X}\mathbf{w}_1)^\top] \\ &= \alpha_2 \mathbb{E}_{p(\mathbf{w}_1)}[\phi(\mathbf{X}\mathbf{w}_1)\phi(\mathbf{X}\mathbf{w}_1)^\top]\end{aligned}$$

Some choices of the activation function ϕ (e.g. cosine, ReLU, tanh) lead to analytical kernels



Same asymptotic behaviour in deep neural networks

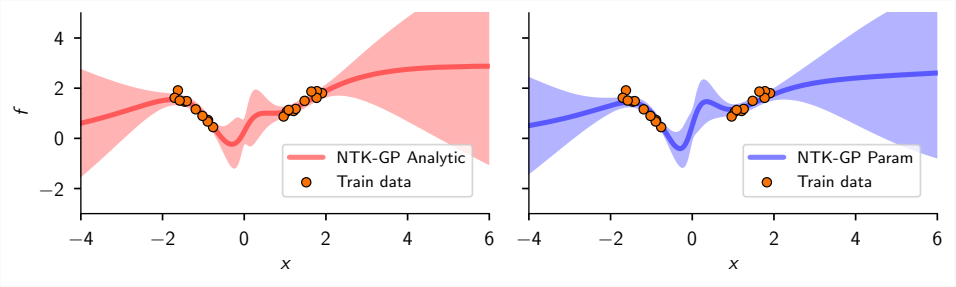
These results are also valid for the deep case:

- Matthews et al. (2018). *Gaussian Process Behaviour in Wide Deep Neural Networks*. ICLR
- Lee et al. (2018). *Deep Neural Networks as Gaussian Processes*. ICLR
- Novak et al. (2019). *Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes*. ICLR
- Garriga-Alonso et al. (2019). *Deep Convolutional Networks as shallow Gaussian Processes*. ICLR
- Yang (2019). *Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes*. NeurIPS
- Khan et al. (2019). *Approximate Inference Turns Deep Networks into Gaussian Processes*. NeurIPS

Other research directions: Neural Tangent Kernel

The evolution of a neural network during training can be described by a kernel (the NTK).

The NTK is random at initialization and evolves during training, but in the infinite-width limit it converges to an explicit limiting kernel and it stays constant during training



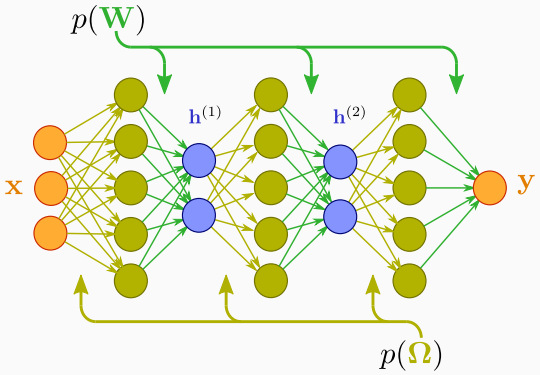
Jacot et al. (2018). *Neural Tangent Kernel: Convergence and Generalization in Neural Networks*. NeurIPS

Lee et al. (2019). *Wide Neural Networks of Any Depth Evolve as Linear Models under Gradient Descent*. NeuIPS

He at al. (2020). *Bayesian Deep Ensembles via the Neural Tangent Kernel*. NeurIPS

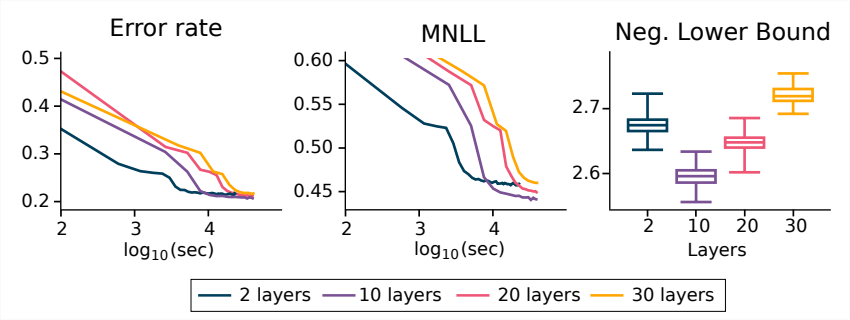
Deep Gaussian Processes and Bayesian Neural Networks

The infinite-limit can be traded-off with compositions of functions:



Damianou and Lawrence (2013). *Deep Gaussian Processes*. AISTATS
Cutajar et al. (2017). *Random Features Expansions for Deep Gaussian Processes*. ICML

Approximate Inference for Deep Gaussian processes



Model selection via a lower bound to the marginal likelihood.

Cutajar et al. (2017). *Random Features Expansions for Deep Gaussian Processes.* ICML

References i

Laplace approximation

- MacKay (1991). *Bayesian Model Comparison and Backprop Nets*. NeurIPS
- MacKay (1991). *A Practical Bayesian Framework for Backpropagation Networks*. Neural comput.
- Williams and Barber (1998). *Bayesian classification with Gaussian processes*. IEEE PAMI
- MacKay (1998). *Choice of Basis for Laplace Approximation*. Machine Learning
- Schraudolph (2002). *Fast curvature matrix-vector products for second-order gradient descent*. Neural Comput.
- Kuss and Rasmussen (2005). *Assessing Approximate Inference for Binary Gaussian Process Classification*. JMLR
- Nickisch and Rasmussen (2008). *Approximations for Binary Gaussian Process Classification*. JMLR
- Martens et al. (2015). *Optimizing Neural Networks with Kronecker-factored Approximate Curvature*. ICML
- Botev et al. (2017). *Practical Gauss-Newton Optimisation for Deep Learning*. ICML
- Ritter et al. (2018). *A Scalable Laplace Approximation for Neural Networks*. ICLR

References ii

- Kunstner et al. (2019). *Limitations of the Empirical Fisher Approximation for Natural Gradient Descent*. NeurIPS
- Dangel et al. (2020). *BackPACK: Packing more into Backprop*. ICLR
- Immer et al. (2021). *Improving predictions of Bayesian neural nets via local linearization*. AISTATS
- Immer et al. (2021). *Scalable Marginal Likelihood Estimation for Model Selection in Deep Learning*. ICML
- Kristiadi et al. (2021). *Learnable Uncertainty under Laplace Approximations*. UAI

Ensemble methods

- Newton and Raftery (1994). *Approximate Bayesian Inference with the Weighted Likelihood Bootstrap*. JRSS - Series B
- Lakshminarayanan et al. (2017). *Simple and scalable predictive uncertainty estimation using deep ensembles*. NeurIPS
- Pearce et al. (2018). *Bayesian Inference with Anchored Ensembles of Neural Networks, and Application to Reinforcement Learning*. ICML Workshop
- Pearce et al. (2018). *Bayesian neural network ensembles*. NeurIPS Workshop

References iii

- Garipov et al. (2018). *Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs*. NeurIPS
- Fort et al. (2019). *Deep Ensembles: A Loss Landscape Perspective*. NeurIPS BDL Workshop
- Milios et al. (2020). *Parametric Bootstrap Ensembles as Variational Inference*. AABI
- He at al. (2020). *Bayesian Deep Ensembles via the Neural Tangent Kernel*. NeurIPS

Infinite-limit Neural Networks

- Rasmussen and Williams (2006). *Gaussian Processes for Machine Learning*, MIT Press
- Damianou and Lawrence (2013). *Deep Gaussian Processes*. AISTATS
- Cutajar et al. (2017). *Random Features Expansions for Deep Gaussian Processes*. ICML
- Jacot et al. (2018). *Neural Tangent Kernel: Convergence and Generalization in Neural Networks*. NeurIPS
- Matthews et al. (2018). *Gaussian Process Behaviour in Wide Deep Neural Networks*. ICLR
- Lee et al. (2018). *Deep Neural Networks as Gaussian Processes*. ICLR
- Novak et al. (2019). *Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes*. ICLR

References iv

- Garriga-Alonso et al. (2019). *Deep Convolutional Networks as shallow Gaussian Processes*. ICLR
- Yang (2019). *Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes*. NeurIPS
- Khan et al. (2019). *Approximate Inference Turns Deep Networks into Gaussian Processes*. NeurIPS
- Lee et al. (2019). *Wide Neural Networks of Any Depth Evolve as Linear Models under Gradient Descent*. NeurIPS