HELMHOLTZ AI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT

Uncertainty quantification for neural network models

Steve Schmerler

Helmholtz AI @HZDR | HZDR ML symposium / 2021-12-06

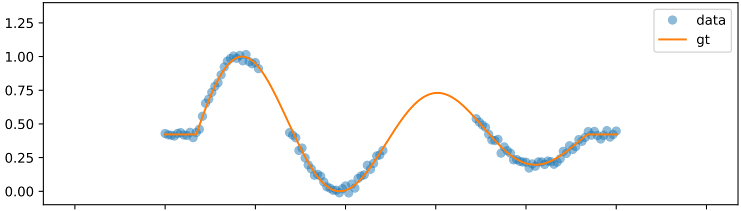
Motivation

- trained ML model f, provide prediction $f(x)$ *plus* "error bar" / model confidence
- uncertainty quantification critical in many neural network (NN) applications (driving, health, ...)
- Helmholtz AI voucher with A. Cangi, L. Fiedler, S. Kulkarni @CASUS

  https://github.com/mala-project

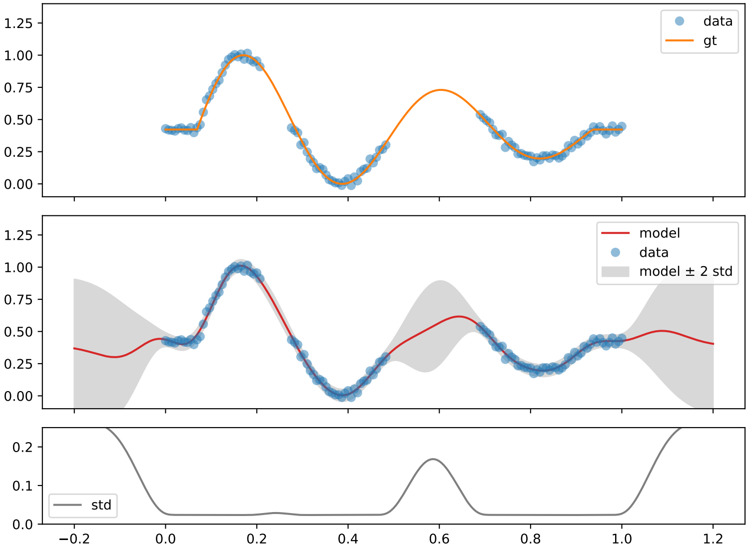
- out-of-distribution detection for NN surrogate models (detect x very dissimilar to training data)
- essential part of an active learning loop

Toy data



- $e^{-\alpha x} \sin(x) +$ Gaussian noise
- add regions of missing data

Gaussian process regression baseline



- GP: Bayesian method that models a Gaussian posterior distribution $\mathcal{N}(f)$ over model functions $f(x)$
- provides predictions $\hat{y}(x) = \mu(x)$ and uncertainty via $\sigma(x)$

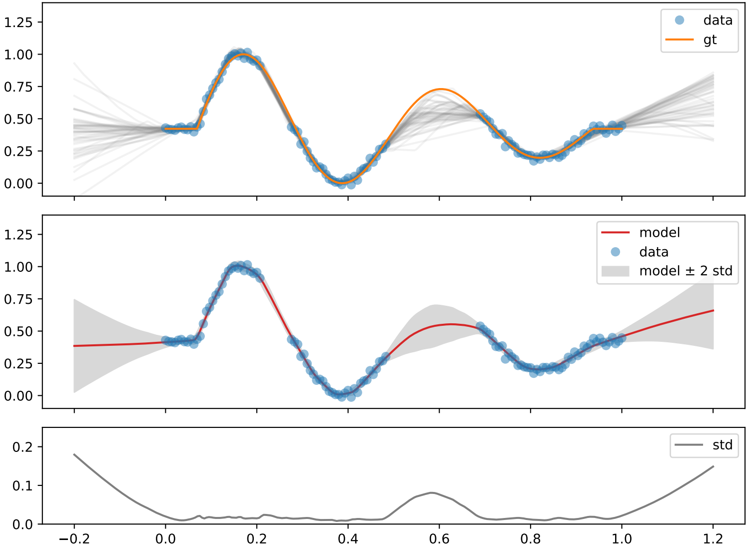
Ensembles

- NN models $f(x;\theta)$ provide point estimates, but no uncertainty
- "sample" $n = 1, \ldots, N$ NNs $f_n(x;\theta_n)$, calculate μ (mean) and σ from N function samples
- different random initialization for θ_n
- train $N\times$
- train each f_n until test loss $\sum_i \|f_n(x_i) - y_i\| < \tau$ with convergence tolerance τ
- provides N models that fit the data equally well (defined by τ) but behave undetermined in out-of-distribution (OOD) data regions
- exploit NN's flexibility in OOD regions ("bad extrapolation")

B. Lakshminarayanan, A. Pritzel, and C. Blundell. "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles". In: *Adv. Neural Inf. Process. Syst.* 30 (2017)

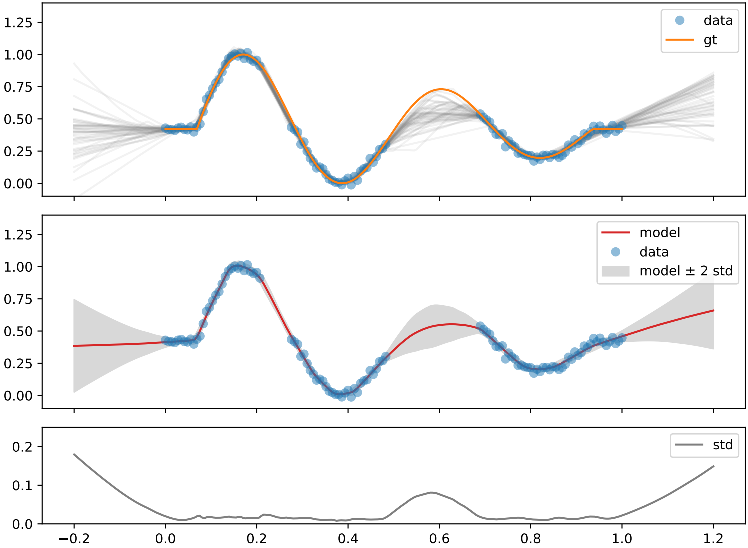
S. Fort, H. Hu, and B. Lakshminarayanan. *Deep Ensembles: A Loss Landscape Perspective.* 2020. URL: http://arxiv.org/abs/1912.02757

Ensembles: $N = 50$



- LeakyReLU, net: 1-100-100-1

Ensembles: $N = 50$

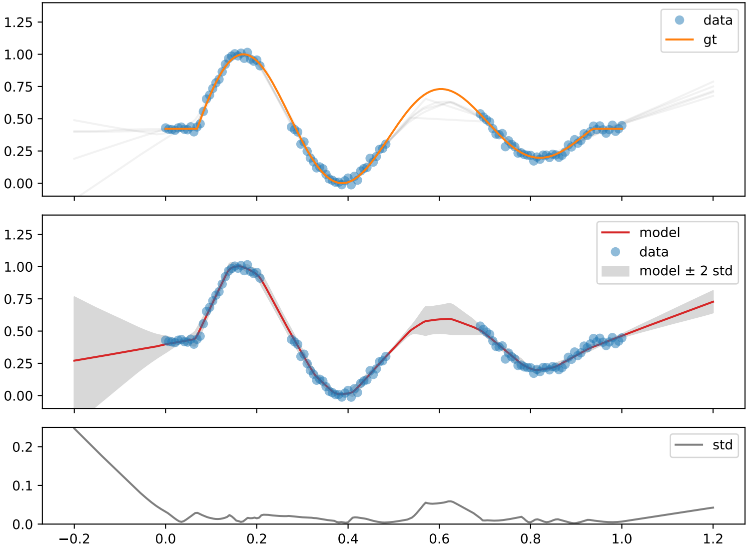


- LeakyReLU, net: 1-100-100-1

- compares well to GP baseline
- works out of the box, no model change needed
- easy to parallelize

- computational cost $N\times$ training time
- prohibitive for large models

Ensembles: $N = 5$



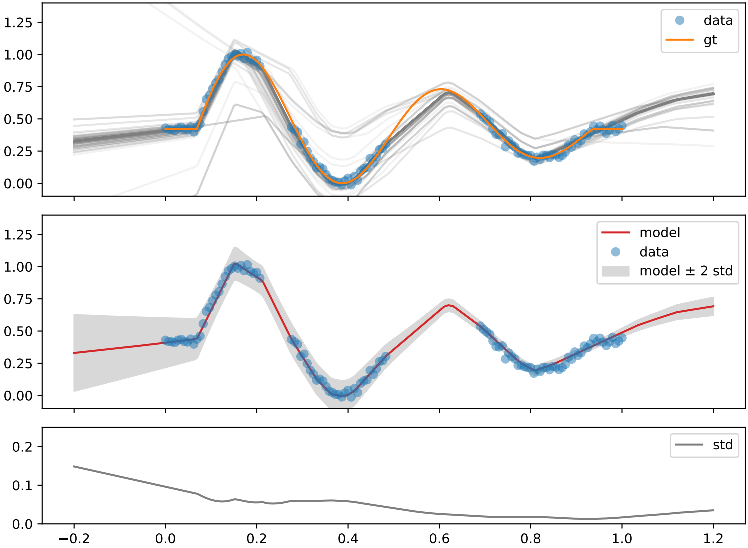
- good news: $N = 5$!
- often tiny ensembles still provide a useful uncertainty estimate

Monte Carlo dropout

- as in ensembles, generate multiple NNs by sampling
- sub-sample 1 trained NN N times using dropout layers
- simple to implement, but need to add dropout layers
- hyper parameter: dropout probability of layer(s), use for calibration

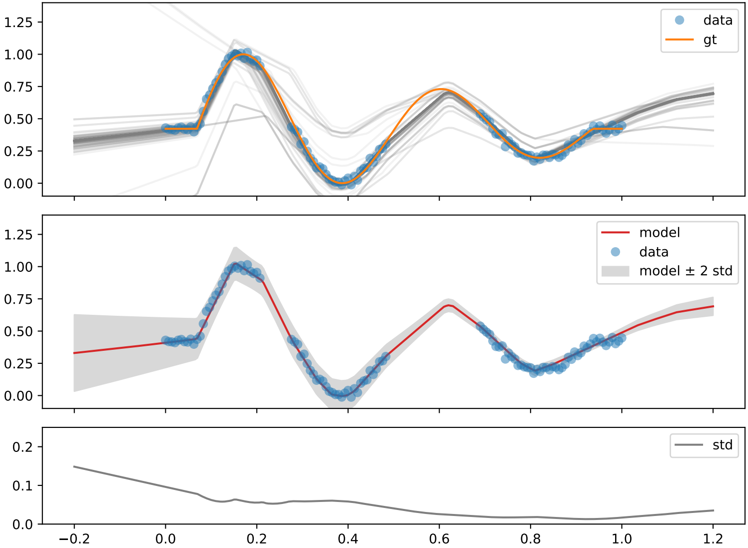
Y. Gal and Z. Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 2016, pp. 1050–1059

Monte Carlo dropout



- LeakyReLU, net: 1-100-100-1

Monte Carlo dropout



- LeakyReLU, net: 1-100-100-1

- very fast, negligible compared to training
- well suited for large models

- sampled models highly correlated $\rightarrow$ stuck with one base model, σ less expressive

Laplace approximation

Apply as post-processing step after training (minimization of loss $\mathcal{L} \in \mathbb{R}$ as function of model parameters $\theta \in \mathbb{R}^D$ where e.g. $\mathcal{O}(D) = 10^6$).

$$\theta^* = \arg\min_\theta \mathcal{L}(\theta)$$

E. Daxberger et al. *Laplace Redux – Effortless Bayesian Deep Learning*. 2021. URL: http://arxiv.org/abs/2106.14806, https://github.com/AlexImmer/Laplace

Laplace approximation

Apply as post-processing step after training (minimization of loss $\mathcal{L} \in \mathbb{R}$ as function of model parameters $\theta \in \mathbb{R}^D$ where e.g. $\mathcal{O}(D) = 10^6$).

$$\theta^* = \arg\min_\theta \mathcal{L}(\theta)$$

Assume $\mathcal{L}$ can be approximated locally at θ^* to second order (with gradient $g = \nabla\mathcal{L}|_{\theta^*}$, Hessian $H = \partial^2\mathcal{L}|_{\theta^*}$ and $h = \theta - \theta^*$) by

$$\mathcal{L}(\theta) \approx \mathcal{L}(\theta^*) + \underbrace{g^\top h}_{=0} + \frac{1}{2} h^\top H h \,.$$

Then one can construct a probability distribution over θ (i.e. over models) such that

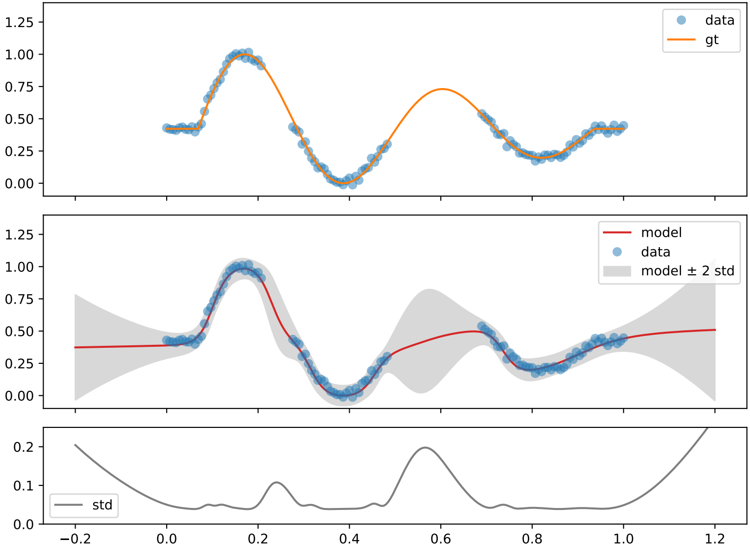
$$p(\theta) \approx \mathcal{N}(\theta|\theta^*, \Sigma)$$

where the covariance matrix is the inverse Hessian

$$\Sigma = H^{-1} \,.$$

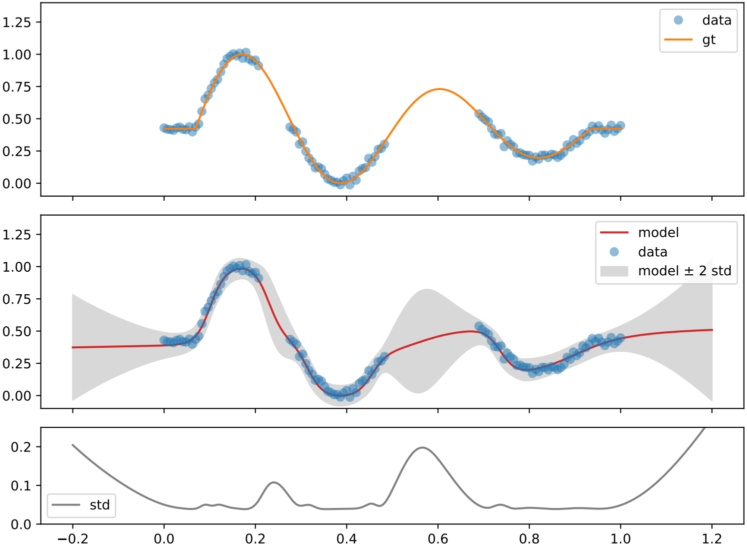
E. Daxberger et al. *Laplace Redux – Effortless Bayesian Deep Learning*. 2021. URL: http://arxiv.org/abs/2106.14806, https://github.com/AlexImmer/Laplace

Laplace approximation



- $\tanh$, net: 1-100-100-1

Laplace approximation



- $\tanh$, net: 1-100-100-1

- no need to change model
- fast, several approximate methods available to scale to large models

- some activations not yet supported by all Hessian approximation schemes (e.g. $\mathrm{LeakyReLU}$ and `kron`)

Summary

- no free lunch, use at least two methods and compare results
- ensembles
  - easy to implement and parallelize
  - few models may be sufficient
  - use as baseline
- Laplace
  - scaling to large models needs approximate Hessian schemes, test carefully
  - quality of results depends on how well the second-order assumption of the local loss holds