NOMU: Neural Optimization-based Model Uncertainty

ICML 2022

Jakob Heiss*, Jakob Weissteiner*, Hanna Wutte*, Sven Seuken, Josef Teichmann

July 10, 2022

What is (good) model uncertainty?

Model Uncertainty (Epistemic Uncertainty) vs Data Noise (Aleatoric Uncertainty)

Bayesian Uncertainty Framework:
$D^{\text{train}} := \{(x_i^{\text{train}}, y_i^{\text{train}})\}$ i.i.d samples from

$$y(x) = f(x) + \varepsilon$$
$$f \sim p_f,\ \varepsilon | x \sim \mathcal{N}(0, \sigma_{\text{n}}^2),$$

Model Uncertainty (Epistemic Uncertainty) vs Data Noise (Aleatoric Uncertainty)

Bayesian Uncertainty Framework:
$D^{\text{train}} := \{(x_i^{\text{train}}, y_i^{\text{train}})\}$ i.i.d samples from

$$y(x) = f(x) + \varepsilon$$
$$f \sim p_f,\ \varepsilon | x \sim \mathcal{N}(0, \sigma_{\text{n}}^2),$$

Posterior for f: $\mathbb{P}[f | D^{\text{train}}]$

"model uncertainty" $\mathbb{V}[f(x) | D^{\text{train}}] =: \sigma_{\text{f}}^2(x)$

Model Uncertainty (Epistemic Uncertainty) vs Data Noise (Aleatoric Uncertainty)

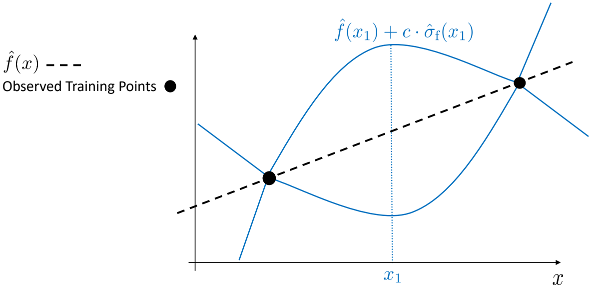
Bayesian Uncertainty Framework:
$D^{\text{train}} := \{(x_i^{\text{train}}, y_i^{\text{train}})\}$ i.i.d samples from

$$y(x) = f(x) + \varepsilon$$
$$f \sim p_f,\ \varepsilon | x \sim \mathcal{N}(0, \sigma_n^2),$$

Posterior for f: $\mathbb{P}[f | D^{\text{train}}]$

"model uncertainty" $\mathbb{V}[f(x) | D^{\text{train}}] =: \sigma_f^2(x)$

No noise: $\varepsilon = 0$



Model Uncertainty (Epistemic Uncertainty) vs Data Noise (Aleatoric Uncertainty)

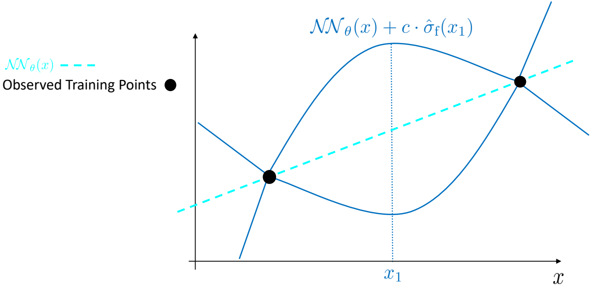
Bayesian Uncertainty Framework:
$D^{\text{train}} := \{(x_i^{\text{train}}, y_i^{\text{train}})\}$ i.i.d samples from

$$y(x) = f(x) + \varepsilon$$
$$f \sim p_f,\ \varepsilon | x \sim \mathcal{N}(0, \sigma_{\text{n}}^2),$$

Posterior for f: $\mathbb{P}[f | D^{\text{train}}]$

"model uncertainty" $\mathbb{V}[f(x)|D^{\text{train}}] =: \sigma_{\text{f}}^2(x)$

No noise: $\varepsilon = 0$



Prior Work. Capturing Model Uncertainty for Neural Networks

- Bayesian neural networks (BNNs)

Prior Work. Capturing Model Uncertainty for Neural Networks

- Bayesian neural networks (BNNs)
    - Markov chain Monte Carlo (MCMC) methods [DJW+20]
    - Variational inference [Gra11, BCKW15, HLA15]
    - Linearized Laplace (LL) [FLHLT19]
    - Neural linear models (NLM) [OR19]

Prior Work. Capturing Model Uncertainty for Neural Networks

- Bayesian neural networks (BNNs) are expensive to train, careful HP tuning needed $\rightarrow$ rarely used in practice [WRV$^+$20]
  - Markov chain Monte Carlo (MCMC) methods [DJW$^+$20]
  - Variational inference [Gra11, BCKW15, HLA15]
  - Linearized Laplace (LL) [FLHLT19]
  - Neural linear models (NLM) [OR19]

Prior Work. Capturing Model Uncertainty for Neural Networks

- Bayesian neural networks (BNNs) are expensive to train, careful HP tuning needed $\rightarrow$ rarely used in practice [WRV+20]
  - Markov chain Monte Carlo (MCMC) methods [DJW+20]
  - Variational inference [Gra11, BCKW15, HLA15]
  - Linearized Laplace (LL) [FLHLT19]
  - Neural linear models (NLM) [OR19]
- Ensemble methods

Prior Work. Capturing Model Uncertainty for Neural Networks

- Bayesian neural networks (BNNs) are expensive to train, careful HP tuning needed $\rightarrow$ rarely used in practice [WRV$^+$20]
  - Markov chain Monte Carlo (MCMC) methods [DJW$^+$20]
  - Variational inference [Gra11, BCKW15, HLA15]
  - Linearized Laplace (LL) [FLHLT19]
  - Neural linear models (NLM) [OR19]
- Ensemble methods
  - Monte Carlo dropout (MCDO) [GG16]
  - Deep ensembles (DE) [LPB17] and hyper deep ensembles (HDE) [WSTJ20]

Prior Work. Capturing Model Uncertainty for Neural Networks

- Bayesian neural networks (BNNs) are expensive to train, careful HP tuning needed $\rightarrow$ rarely used in practice [WRV+20]
  - Markov chain Monte Carlo (MCMC) methods [DJW+20]
  - Variational inference [Gra11, BCKW15, HLA15]
  - Linearized Laplace (LL) [FLHLT19]
  - Neural linear models (NLM) [OR19]
- Ensemble methods often don't satisfy desired properties of model uncertainty!
  - Monte Carlo dropout (MCDO) [GG16]
  - Deep ensembles (DE) [LPB17] and hyper deep ensembles (HDE) [WSTJ20]

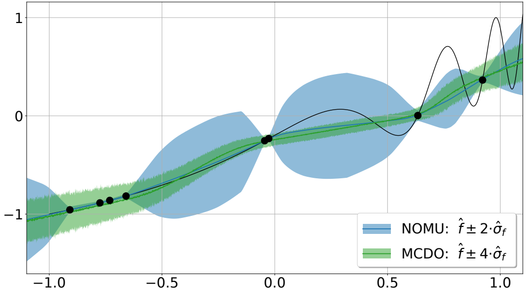
Prior Work. Capturing Model Uncertainty for Neural Networks

- Bayesian neural networks (BNNs) are expensive to train, careful HP tuning needed $\rightarrow$ rarely used in practice [WRV+20]
  - Markov chain Monte Carlo (MCMC) methods [DJW+20]
  - Variational inference [Gra11, BCKW15, HLA15]
  - Linearized Laplace (LL) [FLHLT19]
  - Neural linear models (NLM) [OR19]
- Ensemble methods often don't satisfy desired properties of model uncertainty!
  - Monte Carlo dropout (MCDO) [GG16]
  - Deep ensembles (DE) [LPB17] and hyper deep ensembles (HDE) [WSTJ20]

We introduce NOMU to address these limitations.

Desiderata: What is Good Model Uncertainty (MU)?

D1. Non-negativity of MU, i.e., $\hat{\sigma}_f \geq 0$.

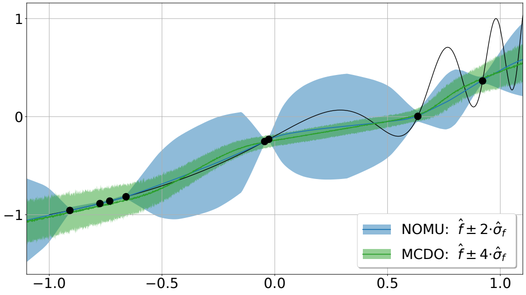


NOMU and a benchmark MC dropout. Shaded areas are MU bounds. The unknown true function is shown as black solid line, training points as black dots.

Desiderata: What is Good Model Uncertainty (MU)?

D1. Non-negativity of MU, i.e., $\hat{\sigma}_f \geq 0$.

D2. Zero MU at training points, i.e., $\hat{\sigma}_f(x^{\text{train}}) = 0$.



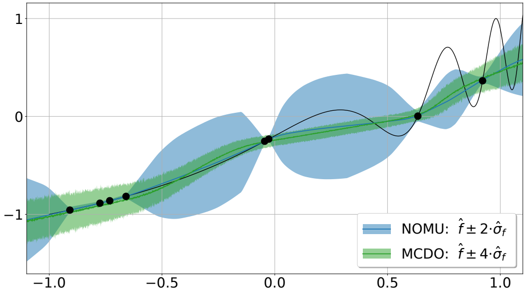
NOMU and a benchmark MC dropout. Shaded areas are MU bounds. The unknown true function is shown as black solid line, training points as black dots.

Desiderata: What is Good Model Uncertainty (MU)?

D1. Non-negativity of MU, i.e., $\hat{\sigma}_f \geq 0$.

D2. Zero MU at training points, i.e., $\hat{\sigma}_f(x^{\text{train}}) = 0$.

D3. Larger MU at points with larger "distance" to training data.



NOMU and a benchmark MC dropout. Shaded areas are MU bounds. The unknown true function is shown as black solid line, training points as black dots.

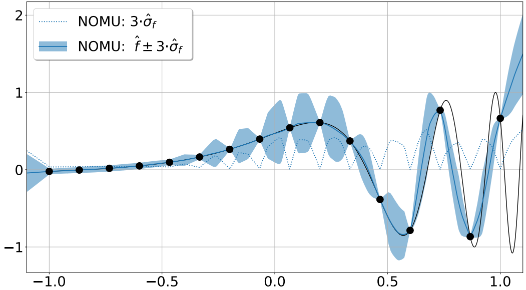
Desiderata: What is Good Model Uncertainty (MU)?

D1. Non-negativity of MU, i.e., $\hat{\sigma}_f \geq 0$.

D2. Zero MU at training points, i.e., $\hat{\sigma}_f(x^{\text{train}}) = 0$.

D3. Larger MU at points with larger "distance" to training data.

D4. Features of x that have high predictive power on the training set have a large effect on the "distance" metric in D3. D4



NOMU fulfills D4. Shaded areas are MU bounds. The unknown true function is shown as black solid line, training points as black dots.

Desiderata: What is Good Model Uncertainty (MU)?

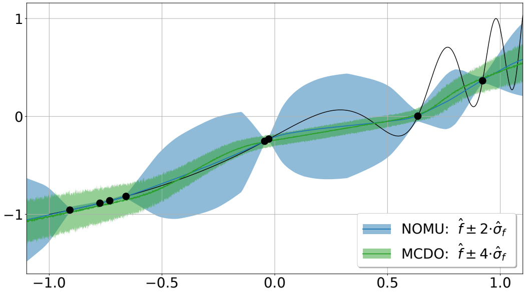
D1. Non-negativity of MU, i.e., $\hat{\sigma}_f \geq 0$.

D2. Zero MU at training points, i.e., $\hat{\sigma}_f(x^{\text{train}}) = 0$.

D3. Larger MU at points with larger "distance" to training data.

D4. Features of x that have high predictive power on the training set have a large effect on the "distance" metric in D3. D4

D5. MU vanishes for training data to infinity, i.e., $\lim_{n^{\text{train}} \to \infty} \hat{\sigma}_f \equiv 0$.



NOMU and a benchmark MC dropout. Shaded areas are MU bounds. The unknown true function is shown as black solid line, training points as black dots.

Desiderata: What is Good Model Uncertainty (MU)?

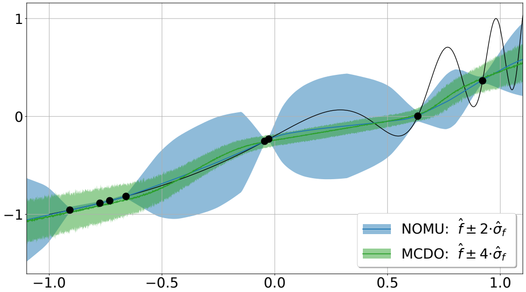
D1. Non-negativity of MU, i.e., $\hat{\sigma}_f \geq 0$.

D2. Zero MU at training points, i.e., $\hat{\sigma}_f(x^{\text{train}}) = 0$.

D3. Larger MU at points with larger "distance" to training data.

D4. Features of x that have high predictive power on the training set have a large effect on the "distance" metric in D3. D4

D5. MU vanishes for training data to infinity, i.e., $\lim_{n^{\text{train}} \to \infty} \hat{\sigma}_f \equiv 0$.



NOMU and a benchmark MC dropout. Shaded areas are MU bounds. The unknown true function is shown as black solid line, training points as black dots.

Many state-of-the-art methods don't satisfy all of these!

Desiderata: What is Good Model Uncertainty (MU)?

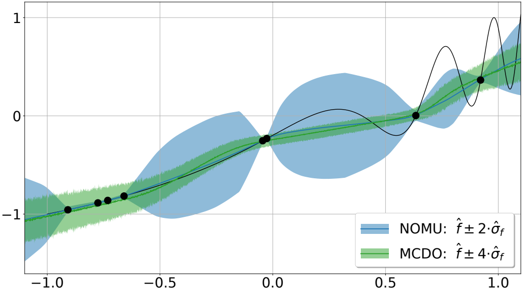
D1. Non-negativity of MU, i.e., $\hat{\sigma}_f \geq 0$. (Prop. D.1.a.)

D2. Zero MU at training points, i.e., $\hat{\sigma}_f(x^{\text{train}}) = 0$. (Prop. D.2.c.)

D3. Larger MU at points with larger "distance" to training data.

D4. Features of x that have high predictive power on the training set have a large effect on the "distance" metric in D3. D4

D5. MU vanishes for training data to infinity, i.e., $\lim_{n^{\text{train}} \to \infty} \hat{\sigma}_f \equiv 0$. (Prop. D.5.a.)



NOMU and a benchmark MC dropout. Shaded areas are MU bounds. The unknown true function is shown as black solid line, training points as black dots.

Many state-of-the-art methods don't satisfy all of these!

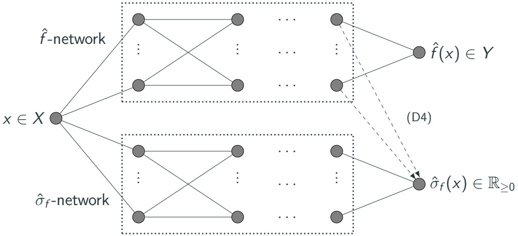
NOMU

NOMU

1. NN-Architecture

2. Loss function (output dim $q = 1$)

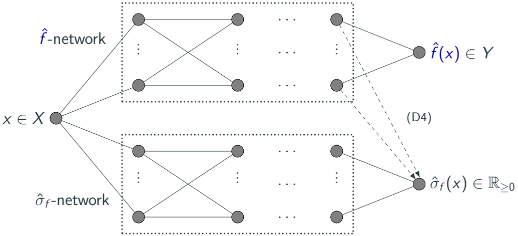
$$\mathcal{NN}_\theta \colon X \to Y \times \mathbb{R}_{\geq 0}$$
$$x \mapsto \mathcal{NN}_\theta(x) := (\hat{f}(x), \hat{\sigma}_f(x))$$



NOMU

1. NN-Architecture

$$\mathcal{NN}_\theta : X \to Y \times \mathbb{R}_{\geq 0}$$
$$x \mapsto \mathcal{NN}_\theta(x) := (\hat{f}(x), \hat{\sigma}_f(x))$$



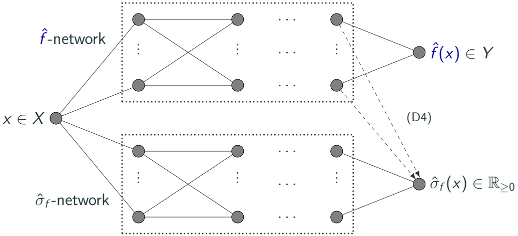
2. Loss function (output dim $q = 1$)

$$L(\mathcal{NN}_\theta) := \underbrace{\sum_{(x,y)\in D^{train}} \ell\left(\hat{f}(x), y\right)}_{\text{standard-loss on } D^{train}}$$

NOMU

1. NN-Architecture

$$\mathcal{N}\mathcal{N}_\theta : X \to Y \times \mathbb{R}_{\geq 0}$$
$$x \mapsto \mathcal{N}\mathcal{N}_\theta(x) := (\hat{f}(x), \hat{\sigma}_f(x))$$



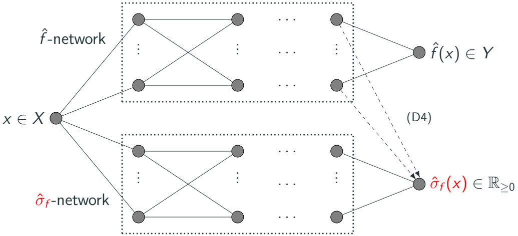
2. Loss function (output dim $q = 1$)

$$L(\mathcal{N}\mathcal{N}_\theta) := \underbrace{\sum_{(x,y)\in D^{train}} \left(\hat{f}(x) - y\right)^2}_{\text{squared-loss on } D^{train}}$$

NOMU

1. NN-Architecture

$$\mathcal{NN}_\theta : X \to Y \times \mathbb{R}_{\geq 0}$$
$$x \mapsto \mathcal{NN}_\theta(x) := (\hat{f}(x), \hat{\sigma}_f(x))$$



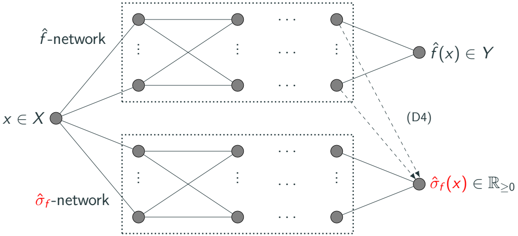
2. Loss function (output dim $q = 1$)

$$L(\mathcal{NN}_\theta) := \underbrace{\sum_{(x,y)\in D^{train}} \left(\hat{f}(x) - y\right)^2}_{\text{squared-loss on } D^{train}} + \mu_{sqr} \cdot \sum_{(x,y)\in D^{train}} (\hat{\sigma}_f(x))^2$$

NOMU

1. NN-Architecture

$$\mathcal{N}\mathcal{N}_\theta : X \to Y \times \mathbb{R}_{\geq 0}$$
$$x \mapsto \mathcal{N}\mathcal{N}_\theta(x) := (\hat{f}(x), \hat{\sigma}_f(x))$$



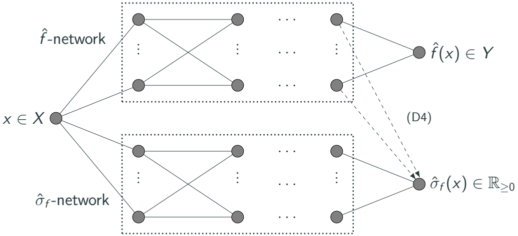
2. Loss function (output dim $q = 1$)

$$L(\mathcal{N}\mathcal{N}_\theta) := \underbrace{\sum_{(x,y)\in D^{train}} \left(\hat{f}(x) - y\right)^2}_{\text{squared-loss on } D^{train}}$$
$$+ \mu_{sqr} \cdot \sum_{(x,y)\in D^{train}} (\hat{\sigma}_f(x))^2$$
$$+ \mu_{exp} \cdot \int_X e^{-c_{exp} \cdot \hat{\sigma}_f(x)} \, dx$$

NOMU

1. NN-Architecture

$$\mathcal{NN}_\theta : X \to Y \times \mathbb{R}_{\geq 0}$$
$$x \mapsto \mathcal{NN}_\theta(x) := (\hat{f}(x), \hat{\sigma}_f(x))$$



2. Loss function (output dim $q = 1$)

$$L(\mathcal{NN}_\theta) := \underbrace{\sum_{(x,y)\in D^{train}} \left(\hat{f}(x) - y\right)^2}_{\text{squared-loss on } D^{train}}$$
$$+ \mu_{sqr} \cdot \sum_{(x,y)\in D^{train}} (\hat{\sigma}_f(x))^2$$
$$+ \mu_{exp} \cdot \int_X e^{-c_{exp} \cdot \hat{\sigma}_f(x)}\, dx$$

- NOMU's Uncertainty bounds: $\hat{f}(x) \pm c\hat{\sigma}_f(x)$, for input $x \in X$.
- NOMU's Hyperparameters: $(\mu_{sqr}, \mu_{exp}, c_{exp})$

Experiments

Regression: BNN Test Bed and UCI Data Sets.

BNN Test Bed: Noiseless Regression.

Table 1: Average NLL and a 95% CI over 200 BNN samples. Winners are marked in gray.

Function	NOMU	GP	MCDO	DE	HDE
BNN1D	-1.65±0.10	-1.08±0.22	-0.34±0.23	-0.38±0.36	8.47±1.00
BNN2D	-1.16±0.05	-0.52±0.11	-0.33±0.13	-0.77±0.07	9.11±0.39
BNN5D	-0.37±0.02	-0.33±0.02	-0.05±0.04	-0.13±0.03	8.41±1.00

Regression: BNN Test Bed and UCI Data Sets.

BNN Test Bed: Noiseless Regression.

Table 1: Average NLL and a 95% CI over 200 BNN samples. Winners are marked in gray.

Function	NOMU	GP	MCDO	DE	HDE
BNN1D	-1.65±0.10	-1.08±0.22	-0.34±0.23	-0.38±0.36	8.47±1.00
BNN2D	-1.16±0.05	-0.52±0.11	-0.33±0.13	-0.77±0.07	9.11±0.39
BNN5D	-0.37±0.02	-0.33±0.02	-0.05±0.04	-0.13±0.03	8.41±1.00

UCI Data Sets: Noisy Regression.

Table 2: Average NLL and a 95% normal-CI over 20 runs for UCI data sets. Winners are marked in gray.

Dataset	NOMU	DE	MCDO	MCDO2	LL	NLM-HPO	NLM
Boston	2.68 ±0.11	2.41 ±0.49	2.46 ±0.11	2.40 ±0.07	2.57 ±0.09	2.58 ±0.17	3.63 ±0.39
Concrete	3.05 ±0.06	3.06 ±0.35	3.04 ±0.03	2.97 ±0.03	3.05 ±0.07	3.11 ±0.09	3.12 ±0.09
Energy	0.77 ±0.06	1.38 ±0.43	1.99 ±0.03	1.72 ±0.01	0.82 ±0.05	0.69 ±0.05	0.69 ±0.05
Kin8nm	-1.08 ±0.01	-1.20 ±0.03	-0.95 ±0.01	-0.97 ±0.00	-1.23 ±0.01	-1.12 ±0.01	-1.13 ±0.01
Naval	-5.63 ±0.39	-5.63 ±0.09	-3.80 ±0.01	-3.91 ±0.01	-6.40 ±0.11	-7.36 ±0.15	-7.35 ±0.01
CCPP	2.79 ±0.01	2.79 ±0.07	2.80 ±0.01	2.79 ±0.01	2.83 ±0.01	2.79 ±0.01	2.79 ±0.01
Protein	2.79 ±0.01	2.83 ±0.03	2.89 ±0.00	2.87 ±0.00	2.89 ±0.00	2.78 ±0.01	2.81 ±0.00
Wine	1.08 ±0.04	0.94 ±0.23	0.93 ±0.01	0.92 ±0.01	0.97 ±0.03	0.96 ±0.01	1.48 ±0.09
Yacht	1.38 ±0.28	1.18 ±0.41	1.55 ±0.05	1.38 ±0.01	1.01 ±0.09	1.17 ±0.13	1.13 ±0.09

Noiseless Bayesian Optimization

- **Goal: maximize an unknown expensive-to-evaluate function.**

Noiseless Bayesian Optimization

- **Goal: maximize an unknown expensive-to-evaluate function.** Good model uncertainty estimate is crucial!

Noiseless Bayesian Optimization

- **Goal: maximize an unknown expensive-to-evaluate function.** Good model uncertainty estimate is crucial!
- Take test functions in 5D-20D.

Noiseless Bayesian Optimization

- **Goal: maximize an unknown expensive-to-evaluate function.** Good model uncertainty estimate is crucial!
- Take test functions in 5D-20D.
- Randomly sample 8 initial points $(x_i, f(x_i))$ and let each algorithm choose 64 further evaluation points (one by one) using its upper UB as acquisition function.

Noiseless Bayesian Optimization

- **Goal: maximize an unknown expensive-to-evaluate function.** Good model uncertainty estimate is crucial!
- Take test functions in 5D-20D.
- Randomly sample 8 initial points $(x_i, f(x_i))$ and let each algorithm choose 64 further evaluation points (one by one) using its upper UB as acquisition function.
- Measure performance based on *final regret*

  $$|\max_{x \in X} f(x) - \max_{i \in \{1,\dots,72\}} f(x_i)| / |\max_{x \in X} f(x)|.$$

Noiseless Bayesian Optimization

- **Goal: maximize an unknown expensive-to-evaluate function.** Good model uncertainty estimate is crucial!
- Take test functions in 5D-20D.
- Randomly sample 8 initial points $(x_i, f(x_i))$ and let each algorithm choose 64 further evaluation points (one by one) using its upper UB as acquisition function.
- Measure performance based on *final regret*

$$|\max_{x \in X} f(x) - \max_{i \in \{1,\dots,72\}} f(x_i)| / |\max_{x \in X} f(x)|.$$

. . . repeat for a number of seeds.

Table 3: BO results: average final regrets per dimension and ranks for each individual function (1=best to 7=worst).

Function	NOMU	GP	MCDO	DE	HDE	pGP	RAND
Levy5D	1	1	6	3	3	4	7
Rosenbrock5D	1	1	1	1	2	5	7
G-Function5D	2	3	1	4	2	3	7
Perm5D	3	1	1	5	7	2	4
BNN5D	1	1	4	1	4	1	7
Average Regret 5D	2.87×10^{-2}	5.03×10^{-2}	4.70×10^{-2}	5.18×10^{-2}	7.13×10^{-2}	4.14×10^{-2}	1.93×10^{-1}
Levy10D	1	3	5	6	1	1	6
Rosenbrock10D	1	1	2	6	3	2	7
G-Function10D	2	5	1	3	2	5	7
Perm10D	2	1	2	6	2	2	1
BNN10D	1	2	1	1	3	1	7
Average Regret 10D	8.40×10^{-2}	1.17×10^{-1}	6.96×10^{-2}	1.15×10^{-1}	9.32×10^{-2}	9.46×10^{-2}	2.35×10^{-1}
Levy20D	1	1	5	7	1	1	6
Rosenbrock20D	2	2	2	6	1	4	6
G-Function20D	1	4	5	1	1	3	7
Perm20D	3	5	3	2	3	3	1
BNN20D	1	2	2	2	6	1	7
Average Regret 20D	1.12×10^{-1}	1.33×10^{-1}	1.39×10^{-1}	1.71×10^{-1}	1.37×10^{-1}	1.17×10^{-1}	2.80×10^{-1}

In a Nutshell

- Capturing model uncertainty for neural networks well is still an open problem.

In a Nutshell

- Capturing model uncertainty for neural networks well is still an open problem.
- We propose NOMU, an optimization based method to characterize model uncertainty.

In a Nutshell

- Capturing model uncertainty for neural networks well is still an open problem.
- We propose NOMU, an optimization based method to characterize model uncertainty.
- NOMU fulfills desirable characteristics of model uncertainty.

In a Nutshell

- Capturing model uncertainty for neural networks well is still an open problem.
- We propose NOMU, an optimization based method to characterize model uncertainty.
- NOMU fulfills desirable characteristics of model uncertainty.
- NOMU performs as well or better than the state-of-the-art in noiseless regression and Bayesian optimization.

In a Nutshell

- Capturing model uncertainty for neural networks well is still an open problem.
- We propose NOMU, an optimization based method to characterize model uncertainty.
- NOMU fulfills desirable characteristics of model uncertainty.
- NOMU performs as well or better than the state-of-the-art in noiseless regression and Bayesian optimization.

Thank you for your time :)

[BCKW15] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. **Weight uncertainty in neural networks.** In *32nd International Conference on Machine Learning (ICML)*, 2015.

[DJW+20] Michael W. Dusenberry, Ghassen Jerfel, Yeming Wen, Yi-An Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. **Efficient and scalable bayesian neural nets with rank-1 factors, 2020.**

[FLHLT19] Andrew YK Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. **"In-between" uncertainty in bayesian neural networks.** *arXiv preprint arXiv:1906.11537*, 2019.

[GG16] Yarin Gal and Zoubin Ghahramani.
Dropout as a bayesian approximation: Representing model uncertainty in deep learning.
In *33rd International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016.

[Gra11] Alex Graves.
Practical variational inference for neural networks.
In *Advances in neural information processing systems*, pages 2348–2356, 2011.

[HLA15] José Miguel Hernández-Lobato and Ryan Adams.
Probabilistic backpropagation for scalable learning of bayesian neural networks.
In *International Conference on Machine Learning*, pages 1861–1869, 2015.

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell.
Simple and scalable predictive uncertainty estimation using deep ensembles.
In *Advances in neural information processing systems*, pages 6402–6413, 2017.

[OR19] Sebastian W Ober and Carl Edward Rasmussen.
Benchmarking the neural linear model for regression.
arXiv preprint arXiv:1912.08416, 2019.

[WRV+20] Florian Wenzel, Kevin Roth, Bastiaan Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin.
How good is the Bayes posterior in deep neural networks really?
In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10248–10259. PMLR, 13–18 Jul 2020.

[WSTJ20] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton.
Hyperparameter ensembles for robustness and uncertainty quantification.

In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

D4 "Metric Learning" Step Function

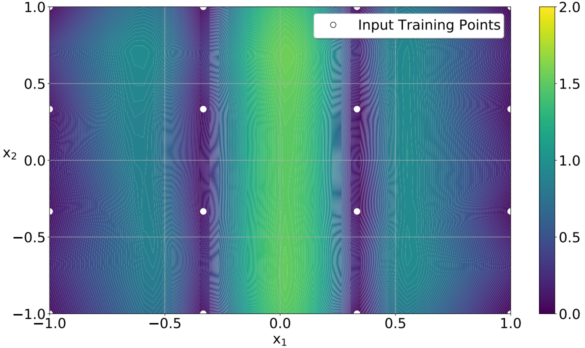
back

- Step function:

$$f = \mathbb{R}^2 \to \mathbb{R} : (x_1, x_2) \mapsto \begin{cases} -1 & \text{if } x_1 < 0 \\ 1 & \text{if } x_1 \geq 0. \end{cases}$$

- Important feature is x_1.
- Output is independent of x_2.

NOMU's model uncertainty.



Bayesian Optimization Motivation