Toward Bayesian Deep Grey-box Modeling

Naoya Takeishi
RCAST, The University of Tokyo
RIKEN Center for Advanced Intelligence Project
ntake@g.ecc.u-tokyo.ac.jp

Abstract

Combining scientific models and deep neural networks (*deep grey-box* or *hybrid modeling*) is expected to be a promising strategy for building robust, partly interpretable, and data-adaptive models. This paper presents a preliminary study to develop a framework for learning deep grey-box models with uncertainty quantification via Bayesian inference of both scientific models and neural nets.

Keywords deep grey-box models · deep hybrid models · Bayesian neural networks

1 Introduction

Despite the success of deep neural networks in a wide range of tasks, their applicability is still limited when extrapolative prediction or interpretation of predictions is required. In contrast, scientific mathematical models are (believed to be) capable of extrapolation and having implications for the modeled phenomena. Meanwhile, such scientific models are sometimes incomplete; they are based on the abstraction of complex real phenomena, which hinders accurate quantitative prediction. In contrast, machine learning models including neural nets are meant to be flexible and can naturally adapt to data. The motivation of *deep grey-box modeling* (or *deep hybrid modeling*) [e.g., 11, 6, 7, 10, 2, 8] is to combine deep neural nets with scientific models to complement the weaknesses of the two.

A technical challenge in deep grey-box modeling is to strike a balance between machine learning and scientific models. The excess flexibility of deep neural nets may work unfavorably in grey-box modeling because they can overwrite the scientific models. For example, consider the additive combination of two models, a neural net f_NN and an incomplete scientific model f_sci, that is, $y \approx f_\mathrm{NN}(x;\theta_\mathrm{NN}) + f_\mathrm{sci}(x;\theta_\mathrm{sci})$ in a task to predict y from x, where θ_NN and θ_sci are the unknown parameters of the neural net and the scientific model, respectively. When f_NN can fit to any function, which is the case with deep neural nets, empirical risk minimization may result in f_NN fitting $y - f_\mathrm{sci}(x;\theta_\mathrm{sci})$ to similarly good extents for any value of θ_sci, meaning choosing a specific value of θ_sci is impossible. This issue happens as well in the general composition of the two models in the form of $f_\mathrm{NN} \circ f_\mathrm{sci}$.[1] We thus need to somehow limit the flexibility of f_NN by regularization; in fact, it has already been addressed [11, 7], at least partly, and is not the main subject of this paper.

We will work on the uncertainty quantification of deep grey-box models, which is also an important step for gaining insights of phenomena from inferred models. While most existing studies on deep grey-box modeling, such as [11, 7], deal with point estimation of the model parameters, we will develop a framework for Bayesian inference of the parameters of both parts of a deep grey-box model, that is, a deep neural net and a scientific model. One of the recent studies most relevant to our motivation is by Akhare *et al.* [1], where they perform the posterior inference of grey-box models using the stochastic weight averaging [5] and deep ensembles [3]. However, they do not consider the regularization of neural nets to limit their flexibility. Since such a regularizer is usually defined on the function value of a neural network, instead of the network's parameters, the problem becomes a kind of generalized Bayesian inference, and thus the inference method should be (often slightly) modified accordingly. In this paper, we report a preliminary study to develop a framework for posterior inference of deep grey-box models that can handle regularization terms to strike a balance between neural nets and scientific models.

[1] The composition in the opposite direction, $f_\mathrm{sci} \circ f_\mathrm{NN}$, is also interesting but out of the scope here because its overall expressiveness is bounded by that of f_sci, and thus the balancing the models becomes much less problematic.

2 Deep grey-box models

Suppose a task to predict y from x for the sake of argument, though the same discussion holds for other kinds of tasks. We are interested in models in the following form of the composition of two functions, f_{NN} and f_{sci}:

$$y \approx f(x; \boldsymbol{\theta}_{\text{NN}}, \boldsymbol{\theta}_{\text{sci}}) = \mathcal{C}\big[f_{\text{NN}}, f_{\text{sci}}; x\big], \tag{1}$$

where $\mathcal{C}$ is a functional that composes the two functions (via, e.g., addition, multiplication, composition, ODE solvers, optimization, etc., and their further compositions). We denote the unknown parameters of the two functions, f_{NN} and f_{sci}, by $\boldsymbol{\theta}_{\text{NN}}$ and $\boldsymbol{\theta}_{\text{sci}}$, respectively. We further suppose that f_{NN} is considerably more flexible than f_{sci}, e.g., f_{NN} is a deep neural network while f_{sci} is a handcrafted (yet scientifically meaningful) mathematical model. Such a composition of functions makes sense when the scientific model, f_{sci}, is somewhat incomplete, e.g., when it suffices as a qualitative approximation of a phenomenon but lacks certain aspects of reality due to the abstraction in model-making. See [8] for more detailed definition of such a model.

Care must be taken in learning deep grey-box models because a deep neural net f_{NN} can, it only, fit the relation from x to y, and the value of f_{sci} may be ignored. Thus we need to strike a balance of the expressivity between the two models, without which grey-box modeling loses its advantages. One may achieve a good balance by meticulously tuning the complexity of f_{NN} by, for example, adjusting the number of layers and units of the network. However, manually tuning the complexity of neural nets by architectural design is hardly feasible. Instead of designing network architecture, a working method to inhibit the excess flexibility of f_{NN} is via constraint or regularization [11, 7, 8]. For example, the method by Yin *et al.* [11] minimizes the norm of f_{NN} for the additive combination of the two models, and Takeishi and Kalousis [7] propose a regularizer that minimizes the effect of f_{NN} for more general combination of the two models.

We will denote such a regularizer by R. Also, we will denote the main loss function (e.g., prediction error) by L. Importantly, R is usually computed with the values of f_{NN} instead of those of the full function, f, because the purpose of R is to minimize the effect of the neural net. For example, it is common to minimize the functional norm of the neural net [11], that is, $R = \Vert f_{\text{NN}} \Vert^2$. Consequently, R usually does not need the information of y, and thus it is possible to compute L and R using different minibatches of x, which is particularly helpful when the amount of data is limited.

3 Particle-based variational inference for Bayesian deep grey-box modeling

Despite the variety of methods for Bayesian inference of neural network parameters, in this work we opt to use the particle-based variational inference (VI) based on the functional space similarity measure [9] because of its controllable flexibility and principled nature. Note that we here do not intend to claim any definitive superiority of the method; our interest does not lie in comparing different methods for Bayesian neural net inference.

3.1 Function space particle-based VI

Suppose a prediction model $g(x; \boldsymbol{\theta}) : \mathcal{X} \to \mathbb{R}$ with input $x \in \mathcal{X}$ and parameters $\boldsymbol{\theta} \in \mathbb{R}^p$. Particle-based VI approximates the posterior distribution of $\boldsymbol{\theta}$ with a set of particles, $\{\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(n)}\}$. The method by Wang *et al.* [9] performs the particle-based VI by measuring the similarity between particles in the function space, instead of the parameter space. Each particle is updated with the following rule:

$$\boldsymbol{\theta}_{\ell+1}^{(i)} \leftarrow \boldsymbol{\theta}_{\ell}^{(i)} - \left(\frac{\partial \boldsymbol{g}_\ell^{(i)}}{\partial \boldsymbol{\theta}_\ell^{(i)}} \right)^\top \boldsymbol{v}\big[\boldsymbol{g}_\ell^{(i)}\big], \tag{2}$$

where $\boldsymbol{\theta}_\ell^{(i)}$ denotes the value of the i-th particle at the ℓ-th iteration, and $\boldsymbol{g}_\ell^{(i)} := [g(x_1; \boldsymbol{\theta}_\ell^{(i)}) \ \cdots \ g(x_m; \boldsymbol{\theta}_\ell^{(i)})]^\top \in \mathbb{R}^m$ is the stack of the model's evaluations (with parameter $\boldsymbol{\theta}_\ell^{(i)}$) on a minibatch of inputs, $X := \{x_1, \dots, x_m\}$.

$\boldsymbol{v} : \mathbb{R}^m \to \mathbb{R}^m$ defines a gradient flow. The gradient flow based on the Stein variational gradient descent [4] is

$$\boldsymbol{v}\big[\boldsymbol{g}_\ell^{(i)}\big] = \frac{1}{n} \sum_{j=1}^{n} \left(k\big(\boldsymbol{g}_\ell^{(i)}, \boldsymbol{g}_\ell^{(j)}\big) \left(\frac{\partial \log p\big(Y \mid \boldsymbol{g}_\ell^{(i)}\big)}{\partial \boldsymbol{g}_\ell^{(i)}} + \frac{\partial \log p\big(\boldsymbol{g}_\ell^{(i)}\big)}{\partial \boldsymbol{g}_\ell^{(i)}} \right) + \frac{\partial k\big(\boldsymbol{g}_\ell^{(i)}, \boldsymbol{g}_\ell^{(j)}\big)}{\partial \boldsymbol{g}_\ell^{(i)}} \right)^\top, \tag{3}$$

where $k\big(\boldsymbol{g}_\ell^{(i)}, \boldsymbol{g}_\ell^{(j)}\big)$ is a kernel function that measures the similarity between two values, $\boldsymbol{g}_\ell^{(i)}$ and $\boldsymbol{g}_\ell^{(j)}$, and Y denotes the set of the labels corresponding to $X = \{x_1, \dots, x_m\}$.

3.2 Applying function space particle-based VI to deep grey-box models

We apply the function space particle-based VI [9] to the deep grey-box model f in eq. (1). We replace the likelihood and the prior distribution in eq. (3) by the loss function L and the regularizer R defined in section 2, respectively. That is, the gradient flow becomes

$$\boldsymbol{v}[\boldsymbol{f}_\ell^{(i)}] = \frac{1}{n}\sum_{j=1}^{n}\left(k\big(\boldsymbol{f}_\ell^{(i)},\boldsymbol{f}_\ell^{(j)}\big)\left(\frac{\partial L}{\partial \boldsymbol{f}_\ell^{(i)}}+\frac{\partial R}{\partial \boldsymbol{f}_\ell^{(i)}}\right)+\frac{\partial k\big(\boldsymbol{f}_\ell^{(i)},\boldsymbol{f}_\ell^{(j)}\big)}{\partial \boldsymbol{f}_\ell^{(i)}}\right)^{\top}. \tag{4}$$

Recall that R is usually computed on the values of f_{NN} and not on those of f. It is thus possible to use different inputs to evaluate R (via f_{NN}'s values) and L (via f's values); it means that there may be two independent computation graphs, $X \to \boldsymbol{f} \to L$ and $X' \to \boldsymbol{f}_{\text{NN}} \to R$, where $X \neq X'$. In this case it is not straightforward to compute $\partial R/\partial \boldsymbol{f}$. We approximate this derivative as follows. By the chain rule, we have

$$\frac{\partial R}{\partial \boldsymbol{f}_\ell^{(i)}}\frac{\partial \boldsymbol{f}_\ell^{(i)}}{\partial \boldsymbol{\theta}} = \frac{\partial R}{\partial \boldsymbol{\theta}}, \tag{5}$$

where $\boldsymbol{\theta} = [\boldsymbol{\theta}_{\text{NN}}^\top \ \boldsymbol{\theta}_{\text{sci}}^\top]^\top$. We compute the minimum norm estimation of $\partial R/\partial \boldsymbol{f}_\ell^{(i)}$, that is,

$$\frac{\partial R}{\partial \boldsymbol{f}_\ell^{(i)}} \approx \frac{\partial R}{\partial \boldsymbol{\theta}}\left(\frac{\partial \boldsymbol{f}_\ell^{(i)}}{\partial \boldsymbol{\theta}}\right)^{\dagger}, \tag{6}$$

where $\cdot^\dagger$ denotes the pseudoinverse of a matrix. Computing eq. (6) using automatic differentiation is straightforward because both R and $\boldsymbol{f}_\ell^{(i)}$ comes after $\boldsymbol{\theta}$ in the computation graph.

3.3 Numerical example

Dataset We used simulated data of a frictionless compound pendulum controlled by some regulator[2]. The original dataset comprises sequences of the pendulum's state, $(\boldsymbol{s}_1, \dots, \boldsymbol{s}_{t_{\text{max}}})$, where $\boldsymbol{s}_t$ is the angle and angular velocity of the pendulum at time t. We collected pairs $(\boldsymbol{x} = \boldsymbol{s}_t, \boldsymbol{y} = (\boldsymbol{s}_{t+\Delta t}, \dots, \boldsymbol{s}_{t+10\Delta t})))$ with $\Delta t = 0.05$, so the task is to predict the pendulum's state up to 10-steps ahead given the current state. We added Gaussian noise with a standard deviation 0.05 as observation noise. We used 3600, 900, and 4500 such pairs as training, validation, and test sets, respectively.

Model We know the equation of motion of an uncontrolled frictionless compound pendulum but do not know the exact formulation of the regulator controlling the pendulum in the dataset. Also, we suppose that the parameters of the equation of motion (the gravity constant g and the length of the pendulum l) are unknown. The grey-box model is

$$\boldsymbol{f}(\boldsymbol{x}) = \text{ODESolve}_{\Delta t,\dots,10\Delta t}\left[\dot{\boldsymbol{s}} = \boldsymbol{f}_{\text{NN}}(\boldsymbol{s};\boldsymbol{\theta}_{\text{NN}}) + \boldsymbol{f}_{\text{sci}}(\boldsymbol{s};\boldsymbol{\theta}_{\text{sci}}) \mid \boldsymbol{s}_0 = \boldsymbol{x}\right], \tag{7}$$

where $\text{ODESolve}_{\Delta t,\dots,10\Delta t}$ denotes the operation to solve the initial value problem in the argument and return the values of the solution evaluated at time steps $t = \Delta t, \dots, 10\Delta t$. $\boldsymbol{f}_{\text{NN}}$ is a neural network with fully connected layers having one hidden layer of size 32. $\boldsymbol{f}_{\text{sci}}$ is from the equation of motion of an uncontrolled frictionless compound pendulum with unknown parameters $\boldsymbol{\theta}_{\text{sci}} = [g \ l]^\top$. We used 50 particles for posterior approximation.

Loss and regularizer The main loss function L is the mean squared error of the prediction, that is, $L = \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{y}\|_2^2$. The regularizer is the norm of $\boldsymbol{f}_{\text{NN}}$, that is, $R = \sum_{\boldsymbol{s}} \boldsymbol{f}_{\text{NN}}(\boldsymbol{s};\boldsymbol{\theta}_{\text{NN}})$. The summation about $\boldsymbol{s}$ was taken using all the values of $\boldsymbol{s}$ that appeared in the computation regarding each minibatch of data.

Results Figure 1 depicts the inferred posterior of $\boldsymbol{\theta}_{\text{sci}} = [g \ l]^\top$. For both elements, the distributions are roughly centered around the data-generating values, $g = 10$ and $l = 1$. The distribution of g happens to have multiple modes, but we do not have enough information so far to discuss if it is something essential or not. Figure 2 is an example of prediction by the learned models. It works as a proof of concept as it seems to capture the uncertainty due to the observation noise, at least to some extent.

[2] Available online at `stable_baselines/gail/dataset/expert_pendulum.npz` of repository `https://github.com/Stable-Baselines-Team/stable-baselines`; retrieved on 19 January 2024.

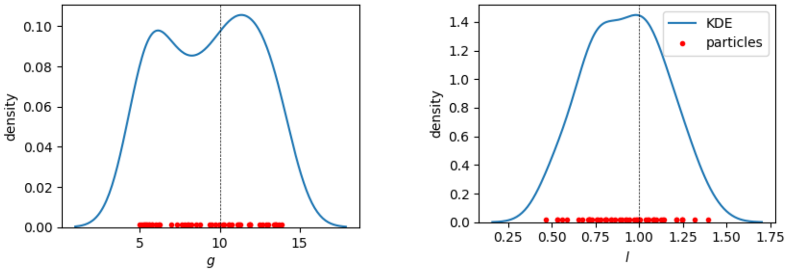

Figure 1: Posterior particles (red dots) of the elements of $\boldsymbol{\theta}_{\text{sci}}$, (*left*) g and (*right*) l. The data-generating values were $g = 10$ and $l = 1$. The densities estimated by the kernel density estimation (blue lines) are present for visualization.

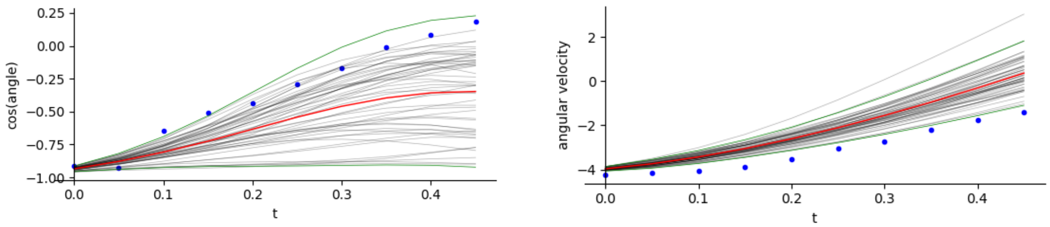

Figure 2: Example of predictions on a test sample. The blue dots are the ground truth; the gray lines show the predictions by each particle; the red lines are the ensemble means; and the green lines indicate the two-sigma areas.

4 Conclusion

We presented a method for Bayesian inference of deep grey-box models. We particularly used the function space particle-based VI [9], for which the gradient of the regularizer should be approximated to avoid the practical difficulty of automatic differentiation. This study is still a work in progress, and there remain several issues toward effective inference for Bayesian deep grey-box modeling. For example, we may want to use different numbers of particles (or different representations) for the posteriors of $\boldsymbol{\theta}_{\text{sci}}$ and $\boldsymbol{\theta}_{\text{NN}}$; as $\boldsymbol{\theta}_{\text{sci}}$ is usually significantly lower dimensional than $\boldsymbol{\theta}_{\text{NN}}$, it may be possible to assign more particles (or more flexible representation) to the former without much sacrificing the computational efficiency. It would be helpful in practice because we are often interested in the detailed shape of the posterior of $\boldsymbol{\theta}_{\text{sci}}$.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP20K19869 and JP21H05300 and JSPS International Joint Research Program Grant Number JPJSJRP-20221501. The author would like to thank Alexandros Kalousis for valuable discussion in the early phase of this work while the author was working at the University of Applied Sciences and Arts Western Switzerland.

References

[1] Deepak Akhare, Tengfei Luo, and Jian-Xun Wang. DiffHybrid-UQ: Uncertainty quantification for differentiable hybrid neural modeling, 2023. arXiv:2401.00161.

[2] Victoriya Kashtanova, Ibrahim Ayed, Andony Arrieula, Mark Potse, Patrick Gallinari, and Maxime Sermesant. Deep learning for model correction in cardiac electrophysiological imaging. In *Proceedings of the 5th International Conference on Medical Imaging with Deep Learning*, pages 665–675, 2022.

[3] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30*, pages 6405–6416, 2017.

[4] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems 29*, pages 2378–2386, 2016.

[5] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems 32*, pages 13153–13164, 2019.

[6] Zhaozhi Qian, William R. Zame, Lucas M. Fleuren, Paul Elbers, and Mihaela van der Schaar. Integrating expert ODEs into neural ODEs: Pharmacology and disease progression. In *Advances in Neural Information Processing Systems 34*, pages 11364–11383, 2021.

[7] Naoya Takeishi and Alexandros Kalousis. Physics-integrated variational autoencoders for robust and interpretable generative modeling. In *Advances in Neural Information Processing Systems 34*, pages 14809–14821, 2021.

[8] Naoya Takeishi and Alexandros Kalousis. Deep grey-box modeling with adaptive data-driven models toward trustworthy estimation of theory-driven models. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, pages 4089–4100, 2023.

[9] Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. Function space particle optimization for Bayesian neural networks. In *Proceedings of the 9th International Conference on Learning Representations*, 2019.

[10] Antoine Wehenkel, Jens Behrmann, Hsiang Hsu, Guillermo Sapiro, Gilles Louppe, and Jörn-Henrik Jacobsen. Robust hybrid learning with expert augmentation. *Transactions of Machine Learning Research*, 2023.

[11] Yuan Yin, Vincent Le Guen, Jérémie Dona, Emmanuel de Bézenac, Ibrahim Ayed, Nicolas Thome, and Patrick Gallinari. Augmenting physical models with deep networks for complex dynamics forecasting. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.