Global Surrogate Modeling by Neural Network-Based Model Uncertainty

Leifur Leifsson[1][0000−0001−5134−870X], Jethro Nagawkar[2][0000−0002−3653−4720], Laurel Barnet[3], Kenneth Bryden[3], Slawomir Koziel[4,5][0000−0002−9063−2647], and Anna Pietrenko-Dabrowska[5][0000−0003−2319−6782]

[1] School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana 47907, USA
leifur@purdue.edu
[2] Department of Aerospace Engineering, Iowa State University, Ames, Iowa 50011, USA
jethro@iastate.edu
[3] Department of Mechanical Engineering, Iowa State University, Ames, Iowa 50011, USA
{labarnet,kmbryden}@iastate.edu
[4] Engineering Optimization & Modeling Center, Department of Engineering, Reykjavík University, Menntavegur 1, 102 Reykjavík, Iceland
koziel@ru.is
[5] Faculty of Electronics Telecommunications and Informatics, Gdansk University of Technology, Narutowicza 11/12, 80-233 Gdansk, Poland
anna.dabrowska@pg.edu.pl

Abstract. This work proposes a novel adaptive global surrogate modeling algorithm which uses two neural networks, one for prediction and the other for the model uncertainty. Specifically, the algorithm proceeds in cycles and adaptively enhances the neural network-based surrogate model by selecting the next sampling points guided by an auxiliary neural network approximation of the spatial error. The proposed algorithm is tested numerically on the one-dimensional Forrester function and the two-dimensional Branin function. The results demonstrate that global surrogate modeling using neural network-based function prediction can be guided efficiently and adaptively using a neural network approximation of the model uncertainty.

Keywords: Global surrogate modeling · neural networks · model uncertainty · error based exploration.

1 Introduction

There is often a need in engineering to assess the performance of a process (e.g., through physical or computer experiments) with a limited number of evaluations. In such cases, surrogate models are often used to approximate the output response of the process over a given data [19, 17, 3]. The surrogates are fast to

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_35

evaluate and can be used to either explore the output response or exploit them to determine a set of parameter values that yield optimal performance.

Modern surrogate modeling strategies start by constructing a surrogate of an initial data set and then progress in cycles using prediction and uncertainty estimates (if available) to select the next sampling point [26]. Several such approaches have been proposed, including the efficient global optimization (EGO) algorithm [9] and Bayesian optimization (BO) [16, 24, 8, 20]. The EGO algorithm [9] follows this strategy by modeling the output response as a random variable and selects the next point to be sampled by maximizing the expected improvement over the best current solution. BO follows the same idea as EGO, but the approach is formalized rigorously through Bayesian theory [16, 24, 20].

Gaussian process regression (GPR) (or Kriging) [11, 7, 3] is widely used with EGO and BO because of its unique feature of providing a prediction of the mean of the underlying data and a prediction of its uncertainty. In particular, GPR provides the mean squared error of the predictor using the same data for constructing the predictor. EGO and BO utilize the predictor and its error estimate to compute a criterion to guide the algorithm to adaptively enhance the predictor. Both EGO and BO typically use ther expected improvement as the criterion [3, 27]. The major disadvantages of GPR modeling, however, are that the computational cost scales cubically with the number of observations, and does not scale well to higher dimensions [13]. This issue can be partially relieved by using graphical processing units (GPUs) and parallel computing [14].

Neural network (NN) regression modeling [6], on the other hand, scales much more efficiently for the optimization of complex and large data sets [13, 22]. It should be noted that the training cost of NNs depends on various factors, such as sample size, number of epochs, and architecture complexity. A major limitation of NN regression modeling is that uncertainty estimates are, in general, not readily available for a single prediction [13]. Rather, it is necessary to make use of an ensemble of NNs with a range of predictions. Bayesian neural networks (BNNs) are an example of such class of algorithms [12, 25, 5]. Current BNN approaches, however, are approximation methods because exact NN-based Bayesian inference is computationally intractable. Using dropout as a Bayesian approximation to represent model uncertainty in deep NNs (DNNs) is an example of one such approach [4]. Current BNN algorithms are, however, computationally intensive.

There is recent interest in creating surrogate modeling algorithms that combine the predictive capabilities of NNs and the uncertainty estimates of GPR. Renganathan et al. [18] use DNNs in place of a polynomial to model the global trend function in GPR modeling. This approach improves the prediction capabilities while still retaining the model uncertainty of GPR. Nevertheless, that approach is still limited in the same way as the original GPR modeling approach. Zhang et al. [28] propose an algorithm that creates and adaptively enhances a multifidelity DNN by exploiting information from low-fidelity data sets. This approach is limited to exploitation only and cannot perform exploration or search a criterion that balances exploration and exploitation.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_35

In this paper, a novel adaptive global surrogate modeling algorithm is proposed that follows the EGO strategy but uses NNs in place of GPR. Specifically, the proposed algorithm iteratively constructs two NN models, one for the prediction of a given process output and the other for the model uncertainty. The proposed algorithm uses separate data sets to construct each NN model. In each cycle, the model uncertainty is used to select the next sampling point and then update the NN prediction model. The algorithm terminates once the uncertainty measure has reached a specified tolerance or the maximum number of samples is reached. In this work, the spatial error in the prediction is used as the uncertainty measure, and it is maximized in each cycle to select the next sampling point. The proposed algorithm is tested on two low-dimensional analytical problems. The results demonstrate that global modeling using NN-based function prediction can be guided efficiently and adaptively by an NN approximation of the model uncertainty.

The next section introduces the proposed algorithm. The following section presents results of numerical experiments using one- and two-dimensional analytical functions. Finally, concluding remarks are presented.

2 Methods

The proposed approach is summarized in Algorithm 1. The algorithm requires two initial data sets that are used to fit separate neural networks. One neural network models the process output in terms of the input parameters, and the other models the spatial error in the first neural network. Let $(\mathbf{X}, \mathbf{Y})_f$ be the set of sample points used to fit the neural network to the process output, and let $(\mathbf{X}, \mathbf{Y})_u$ be the set of sample points used to fit the neural network to the spatial uncertainty. Here, $\mathbf{X}_f = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(p)}\}^T$ is the set of the input parameter sample points and $\mathbf{Y}_f = (y^{(1)}(\mathbf{x}^{(1)}), ..., y^{(p)}(\mathbf{x}^{(p)}))^T$ the corresponding set of model outputs. Furthermore, $\mathbf{X}_u = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(q)}\}^T$ is the set of input parameter sample points and $\mathbf{Y}_u = (y^{(1)}(\mathbf{x}^{(1)}), ..., y^{(q)}(\mathbf{x}^{(q)}))^T$ the corresponding set of model outputs. In this work, it is assumed that the data sets $(\mathbf{X}, \mathbf{Y})_f$ and $(\mathbf{X}, \mathbf{Y})_u$ are distinctly different. Both sets are created using Latin hypercube sampling (LHS) [15].

To fit the neural networks within the proposed algorithm, the mean squared error (MSE) loss function is minimized:

$$\mathcal{L} = \frac{\sum_{l=1}^{N} (\hat{y}^{(l)} - y^{(l)})^2}{N}, \tag{1}$$

where N is the number of samples in the training data. The loss function minimizes the mismatch between the training data, y, and the predicted values, $\hat{y}$, of the neural network [6, 21]. To minimize the loss function, the adaptive moments (ADAM) optimization algorithm is used [10] along with the backpropagation algorithm [2] to compute the gradients. The neural network setup used in this work, in particular the number of hidden layers and the number of neurons per hidden layer, is case dependent and is described in the numerical experiments.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_35

Algorithm 1 Adaptive global surrogate modeling algorithm with neural network-based prediction and uncertainty

Require: initial data sets $(\mathbf{X}, \mathbf{Y})_f$ and $(\mathbf{X}, \mathbf{Y})_u$
  repeat
    fit neural network to function with available data $(\mathbf{X}, \mathbf{Y})_f$
    compute uncertainty with available data $(\mathbf{X}, \mathbf{Y})_u$
    fit neural network to uncertainty with available data $(\mathbf{X}, \mathbf{Y})_u$
    $\mathbf{P} \leftarrow \arg\max_{\mathbf{x}} \hat{s}^2(\mathbf{x})$
    $\mathbf{X}_f \leftarrow \mathbf{X}_f \cup \mathbf{P}$
    $\mathbf{Y}_f \leftarrow \mathbf{Y}_f \cup y(\mathbf{P})$
  until convergence

Other hyperparameters are common between the cases. Specifically, the tangent hyperbolic is used as the activation function, the learning rate is set to 0.001, and the number of epochs is fixed with a value of $3,000$. The neural network algorithm is implemented using TensorFlow [1].

The neural network algorithm is used in each cycle to construct a surrogate model, $\hat{y}_f$, of the process output, y, using $(\mathbf{X}, \mathbf{Y})_f$. In this work, the uncertainty measure of $\hat{y}_f$ is estimated by the square of the spatial error and is written as

$$s^2(\mathbf{x}) = (\hat{y}_f(\mathbf{x}) - y_f(\mathbf{x}))^2. \tag{2}$$

In the proposed algorithm, $s(\mathbf{x})^2$ is computed in each cycle using the data set $(\mathbf{X}, \mathbf{Y})_u$, and the neural network algorithm is used to construct the surrogate model $\hat{s}(\mathbf{x})^2$.

To select the next sampling point in each cycle of the proposed algorithm, the uncertainty measure $\hat{s}(\mathbf{x})^2$ is maximized using differential evolution [23]. The algorithm is terminated if $\hat{s}(\mathbf{x})^2$ is lower than a specified tolerance or the number of cycles exceeds a specified maximum value.

3 Numerical Experiments

The results of numerical experiments with the proposed algorithm are presented in this section. Two analytical cases are considered, the first case has one input parameter and the second has two.

3.1 One-dimensional Forrester Function

The one-dimensional analytical function developed by Forrester et al. is written as

$$y(x) = (6x - 2)^2 \sin(12x - 4), \tag{3}$$

where $x \in [0,1]$. The proposed algorithm is applied to the modeling of this function using three uniformly distributed initial samples and ten infill points.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_35

The total number of samples for modeling the function is, therefore, 13. Ten uniformly distributed samples are used for modeling the uncertainty. The number of hidden layers is set to three and the number of neurons in each hidden layer is set to 50.

Figure 1 shows the modeling progression at iterations 0 (3 initial samples and the first infill point), 3 (3 initial and 3 prior infills and the new infill point), 6 (3 initial and 6 prior infills and the new infill), and 9 (with all the initial and infill samples), respectively. Specifically, each subfigure shows the neural network prediction $\hat{y}(x)$ along with the true function $y(x)$, sample points and the next sampling point, as well as the neural network model of the uncertainty $\hat{s}^2(x)$. The location of the maximum uncertainty in the interval from 0 to 1 guides the sampling so that the prediction quickly aligns with the true function through exploration.

The global accuracy of the surrogate models is measured using the root mean squared error (RMSE), which is evaluated using a separate testing data set. Figure 2(a) shows how the RMSE changes over the the iterations and is reduced to around 0.1. Figure 2(b) shows how the maximum model uncertainty reduces over the iterations from around 250 to 0.01, or by four orders of magnitude.

3.2 Two-dimensional Branin Function

The two-dimensional Branin function is written as

$$y(x_1, x_2) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos(x_1) + 10, \qquad (4)$$

where $x_1, x_2 \in [0, 10]$. The proposed algorithm models this function with ten initial samples selected using LHS and fifty additional infill points for a total of sixty points at the end of fifty iterations. One hundred points, selected through LHS, are used for the uncertainty model. For this case, the number of hidden layers was set to two, with fifty neurons in each hidden layer.

The modeling progression, at iterations 0 (initial samples only), 14, 31, and 49, is shown in Fig. 3. The left plot in each subfigure shows the neural network model of the function $\hat{y}(x_1, x_2)$ with the sample points used and the next selected infill point, while the right plot shows the neural network model of the uncertainty $\hat{s}^2(x_1, x_2)$ that is being used to select that infill point.

Figure 4 illustrates the global improvement of the surrogate model as the algorithm progresses through the iterations with Fig. 4(a) showing how the RMSE for the surrogate model reduces down to 0.2, and Fig. 4(b) showing how the maximum predicted model uncertainty $\hat{s}^2(x_1, x_2)$ reduces by six orders of magnitude (from roughly $3.7 \cdot 10^5$ to 0.1). The close comparison between the final neural network model and the true function can be seen in Fig. 5 with (a) the contour plot of the true function and (b) the final predicted model with all of the sample points indicated.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_35

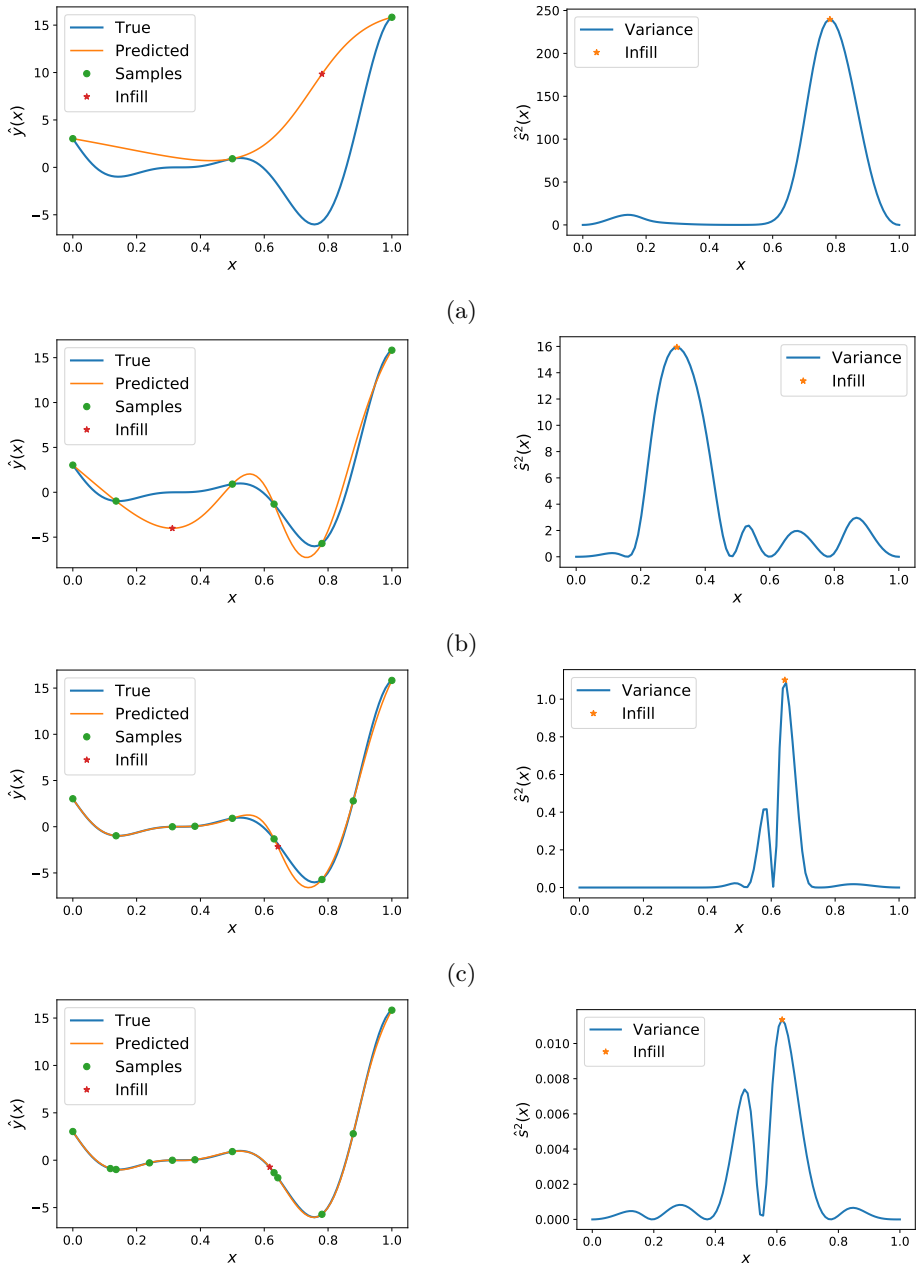


Fig. 1: Forrester function prediction (left) and uncertainty (right) at iterations: (a) 0, (b) 3, (c) 6, (d) 9.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_35

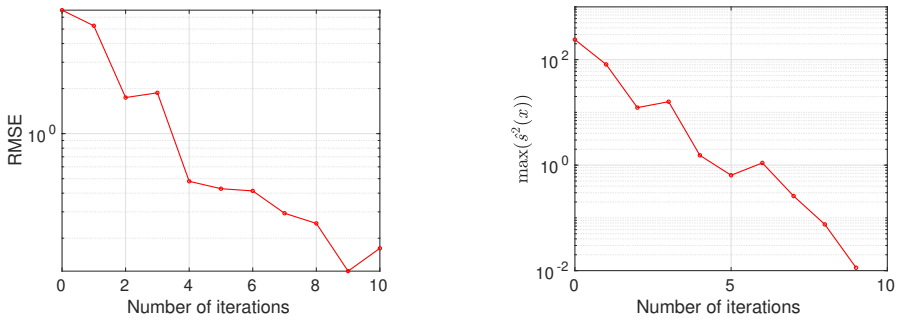


Fig. 2: Forrester function modeling error evolution: (a) root mean squared error of the prediction model, (b) maximum variance of the uncertainty model.

4 Conclusion

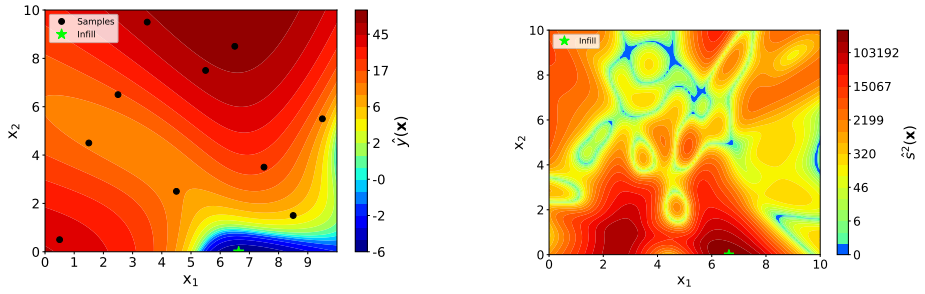
Global modeling of large data sets is important for decision-making in experimentally and computationally-driven discoveries in engineering and science. The proposed approach of combining efficient global optimization strategies and neural network modeling directly tackles this important problem. Specifically, this paper demonstrates that global modeling using neural network-based function prediction can be guided by an auxiliary neural network approximation of the prediction spatial error that enables efficient adaptive surrogate modeling of large data sets. This capability will support scientists and engineers to make decisions on whether and where in the parameter space to do a physical experiment or computational simulation.

Future work will focus on improving the proposed algorithm to permit adaptive sampling of the uncertainty model as well as using data from multiple levels of fidelity. Furthermore, the process of updating the neural network fit in each cycle of the algorithm needs to be accelerated. Other uncertainty metrics than the prediction variance also need to be explored. An important step will be to compare the proposed approach against current state-of-the-art methods and to characterize the computational costs of each approach. Performing numerical experiments on high-dimensional problems involving physical and computational data is of current interest.

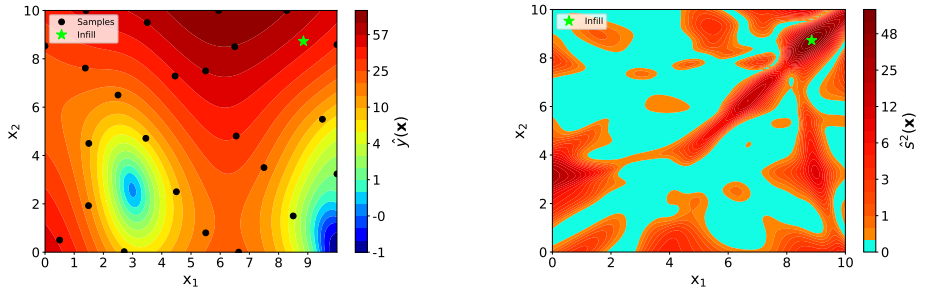
Acknowledgements

This material is based upon work supported in part by the Department of Energy under a Laboratory Directed Research and Development grant at Ames Laboratory.

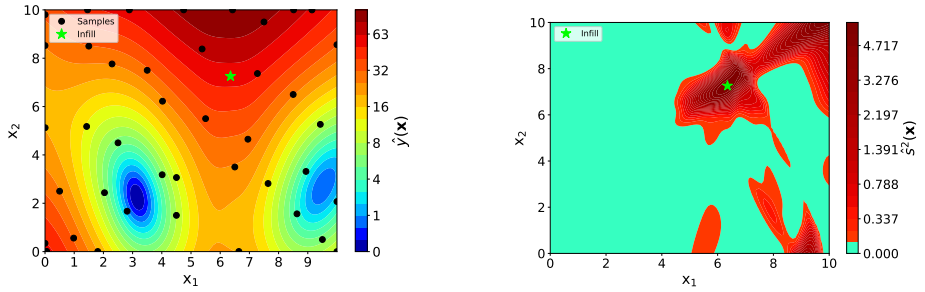
ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_35



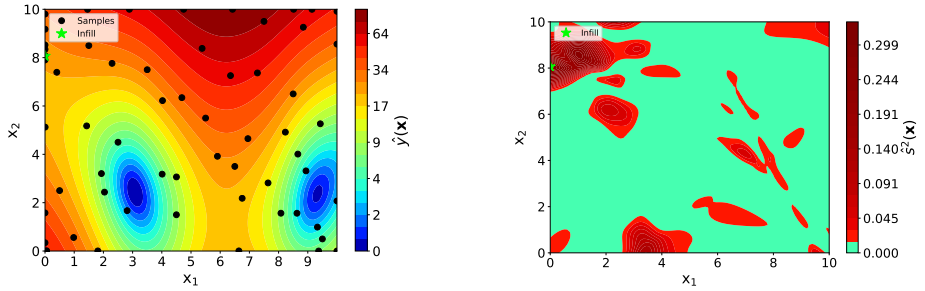
(a)



(b)



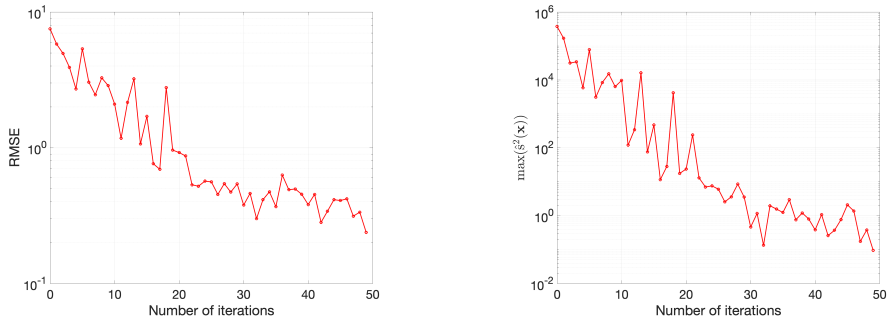
(c)



(d)

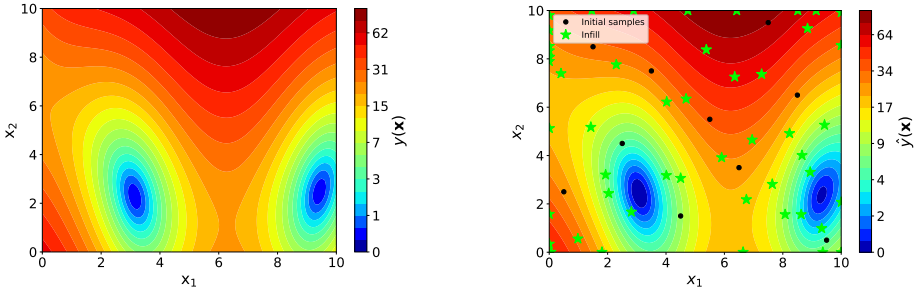
Fig. 3: Branin function prediction (left) and uncertainty (right) at iterations: (a) 0, (b) 14, (c) 31, (d) 49.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_35



(a) (b)

Fig. 4: Branin function global modeling error evolution: (a) root mean squared error of the prediction model, (b) maximum variance of the uncertainty model.



(a) (b)

Fig. 5: Branin function: (a) true, (b) predicted.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_35

References

1. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), http://tensorflow.org/, software available from tensorflow.org
2. Chauvin, Y., Rumelhart, D.E.: Backpropagation: theory, architectures, and applications. Psychology press, Hillsdale, NJ (1995)
3. Forrester, A.I.J., Keane, A.J.: Recent Advances in Surrogate-Based Optimization. Progress in Aerospace Sciences **45**(1-3), 50–79 (2009)
4. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: Proceedings of the 33rd International Conference on Machine Learning. pp. 1050–1059 (2016)
5. Goan, E., Fookes, C.: Bayesian neural networks: An introduction and survey. In: Mengersen, K., Pudlo, P., Robert, C. (eds) Case Studies in Applied Bayesian Data Science. Lecture Notes in Mathematics. pp. 45–87 (2020)
6. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. The MIT Press, Cambridge, MA (2016)
7. Huang, D., Allen, T., Notz, W., Zeng, N.: Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models. Journal of Global Optimization **34**(3), 441–466 (2006)
8. Jones, D.R.: A Non-Myopic Utility Function for Statistical Global Optimization. Journal of Global Optimization **21**, 345–383 (2001)
9. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization **13**(4), 455–492 (1998)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
11. Krige, D.G.: Statistical approach to some basic mine valuation problems on the witwatersrand. Journal of the Chemical, Metallurgical and Mining Engineering Society of South Africa **52**(6), 119–139 (1951)
12. Lampinen, J., Vehtari, A.: Bayesian approach for neural networks - review and case studies. Neural Networks **14**(3), 257–274 (2001)
13. Lim, Y.F., Ng, C.K., Vaitesswar, U.S., Hippalgaonkar, K.: Extrapolative Bayesian Optimization with Gaussain Process and Neural Network Ensemble Surrogate Models. Advanced Intelligent Sytems **3**, 2100101 (2021)
14. Liu, H., Ong, Y.S., Shen, X., Cai, J.: When Gaussian Process Meets Big Data: A Review of Scalable GPs. IEEE Transactions on Neural Networks and Learning Systems **31**(11), 4405–4423 (2020)
15. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2), 239–245 (1979)
16. Mockus, J.: Application of Bayesian Approach to Numerical Methods of Global and Stochastic Optimization. Journal of Global Optimization **4**(4), 347–365 (1994)
17. Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., Tucker, P.K.: Surrogate-based analysis and optimization. Progress in Aerospace Sciences **21**(1), 1–28 (2005)
18. Renganathan, S.A., Maulik, R., Ahuja, J.: Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. Aerospace Science and Technology **111**, 106522 (2021)
19. Sacks, J., Welch, W., Michell, J.T., Wynn, P.H.: Design and Analysis of Computer Experiments. Statistical Science **4**, 409–423 (1989). https://doi.org/10.1214/ss/1177012413

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_35

20. Sasena, M.J.: Flexibility and Efficiency Enhancement for Constrained Global Design Optimization with Kriging Approximations. PhD Thesis, University of Michigan, USA (2002)
21. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks **61**, 85 – 117 (2015). https://doi.org/10.1016/j.neunet.2014.09.003
22. Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M.A., Adams, R.P.: Scalable bayesian optimization using deep neural networks. In: Proceedings of the 32nd International Conference on Machine Learning. pp. 2171–2180 (2015)
23. Storn, R., Price, K.: Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization **11**, 341–359 (1997)
24. Streltsov, S., Vakili, P.: A Non-Myopic Utility Function for Statistical Global Optimization. Journal of Global Optimization **14**(3), 283–298 (1999)
25. Titterington, D.M.: Bayeisan methods for neural networks and related models. Statistical Science **19**(1), 128–139 (2004)
26. Viana, F.A., Haftka, R.T., Watson, L.T.: Efficient global optimization algorithm assisted by multiple surrogate techniques. Journal of Global Optimization **56**, 669–689 (2013)
27. Zhan, D., Xing, H.: Expected improvement for expensive optimization: a review. Journal of Global Optimization **78**, 507–544 (2020)
28. Zhang, X., Xie, F., Ji, T., Zhu, Z., Zheng, Y.: Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. Computer Methods in Applied Mechanics and Engineering **373**(1), 113485 (2021)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_35