Parameter estimation of partial differential equations via neural networks

Alexander Glushko, Dmitry I. Kabanov

July 16, 2019

In this work, we partially reproduce and extend the work of Raissi, Perdikaris, and Karniadakis [2] on applying neural networks to the problem of parameter estimation of partial differential equations from given observations.

Problem setup

We are given data $\boldsymbol{D} = \{t_i, x_i, u_i\}$, $i = 1, \dots, N$, that are observed from the solution of partial differential equation of the form

$$u_t + \mathscr{N}(u; \boldsymbol{\lambda}) = 0, \tag{1}$$

where $u = u(x,t)$ is the solution of the equation, $\mathscr{N}(u; \boldsymbol{\lambda})$ a nonlinear algebraic-differential operator, $\boldsymbol{\lambda}$ the vector of unknown parameters, subscript t denotes differentiation with respect to time. The solution $u(x,t)$ is given on the subset of $\mathbb{R} \times \mathbb{R}$ and the collected observations are distributed uniformly on this subset. The goal is to estimate $\boldsymbol{\lambda}$ from the observations $\boldsymbol{D}$.

By Bayes' rule, the optimal value of $\boldsymbol{\lambda}$ is found through maximization of the posterior distribution [3]

$$\rho(\boldsymbol{\lambda} | \boldsymbol{D}) \propto \rho(\boldsymbol{D} | \boldsymbol{\lambda}) \times \rho(\boldsymbol{\lambda}). \tag{2}$$

We assume that the observed data are given by the exact solution plus some additive uncorrelated Gaussian noise

$$u_i = u(x_i, t_i; \boldsymbol{\lambda}) + \epsilon_i, \quad i = 1, \dots, N, \tag{3}$$

where $\epsilon_i \sim N(0, \sigma^2)$, and we emphasize dependence of the exact solution $u(x, t; \boldsymbol{\lambda})$ on the parameter vector $\boldsymbol{\lambda}$. Furthermore, we assign uninformative prior for $\boldsymbol{\lambda}$

$$\rho(\boldsymbol{\lambda}) = \text{const} \quad \text{for all } \boldsymbol{\lambda}, \tag{4}$$

so that, in principle, the problem of finding $\boldsymbol{\lambda}$ is a nonlinear unconstrained optimization problem

$$\underset{\boldsymbol{\lambda}}{\arg\min} \quad \log \sum_{i=1}^{N} \left[u_i - u(x_i, t_i; \boldsymbol{\lambda})\right]^2, \tag{5}$$

where we marginalized the likelihood treating noise variance σ^2 as a nuisance parameter [3, section 8.2]. In the following we omit log function as its monotonic and we are interested in the optimal value of $\boldsymbol{\lambda}$, not the value of the objective function.

Solution of the optimization problem (5) requires multiple solutions of Eq. (1), which can be expensive. To alleviate this, we follow an approach proposed in [2]. We replace the exact model $u(x_i, t_i; \boldsymbol{\lambda})$ with a surrogate, where the surrogate model is a feedforward neural network [1]

$$u_{\text{NN}}(x, t; \boldsymbol{\theta}) = g_L \circ g_{L-1} \circ \cdots \circ g_1, \tag{6}$$

where

$$g_\ell(z; \boldsymbol{\theta}_\ell) = \sigma(W_\ell z + b_\ell), \quad \ell = 1, \dots, L,$$

with L is the number of layers in the network, with layers 0 and L being input and output layers, respectively, and layers from 1 to $L-1$ being hidden layers, σ being a nonlinear activation function applied componentwise. In this work, we plan to use $\sigma(z) = \tanh(z)$. The neural-network parameter $\boldsymbol{\theta}$ contains the components of matrices $W_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ and bias vectors $b_\ell \in \mathbb{R}^{n_\ell}$, where n_ℓ denotes the width of the ℓ^{th} layer. The hyperparameters of the neural network, L (the number of the layers) and the width of each layer, are to be determined later during the course of the project.

We assume that the discrepancy between the exact solution u and the neural-network approximation u_{NN} is a Gaussian noise, such that the likelihood function is not affected. Furthermore, to ensure that $u_{\text{NN}}(x, t; \boldsymbol{\theta})$ is close to the exact solution of Eq. (1), we, strictly speaking, are supposed to formulate the problem of estimating of the unknown parameters $\boldsymbol{\lambda}$ as the following constrained optimization problem:

$$\underset{\boldsymbol{\lambda}, \boldsymbol{\theta}}{\arg\min} \quad \sum_{i=1}^{N} \left[u_i - u_{\text{NN}}(x_i, t_i; \boldsymbol{\theta})\right]^2 \tag{7a}$$

$$\text{subject to} \quad u_{\text{NN},t} + \mathcal{N}(u_{\text{NN}}; \boldsymbol{\lambda}) = 0. \tag{7b}$$

However, the equality constraint is difficult to satisfy exactly as u_{NN} is just an approximation of the exact solution of Eq. (1). Therefore, we relax the optimization problem in the following way. We introduce a secondary neural network

$$f_{\text{NN}}(x, t; \boldsymbol{\lambda}, \boldsymbol{\theta}) = u_{\text{NN},t} + \mathcal{N}(u_{\text{NN}}; \boldsymbol{\lambda}), \tag{8}$$

where we plug the neural network u_{NN} into Eq. (1) and require that $f_{\text{NN}} \approx 0$. Then both networks are trained simultaneously (that is, their parameters are found) from the observations $\boldsymbol{D}$ via the relaxed version of the optimization problem (7):

$$\underset{\boldsymbol{\lambda}, \boldsymbol{\theta}}{\arg\min} \sum_{i=1}^{N} \left[u_i - u_{\text{NN}}(x_i, t_i; \boldsymbol{\theta})\right]^2 + \gamma \sum_{i=1}^{N} \left[f_{\text{NN}}(x_i, t_i; \boldsymbol{\lambda}, \boldsymbol{\theta})\right]^2, \tag{9}$$

where γ is an extra hyperparameter that controls the importance of the penalty term. This parameter is chosen via cross-validation. We assume that the number of observations N is fixed (not controlled by us) and we choose the size of the network empirically by varying its depth until satisfying agreement is found.

Uncertainty quantification

Training of neural networks in [2] lacks uncertainty quantification for the found parameters, that is, only the point estimates are provided. In this work, we apply the bootstrap procedure [4] to simulate $\boldsymbol{\lambda}$ and estimate its confidence sets.

Applications

1. As a concrete example, we consider the linear heat equation

$$u_t - \lambda u_{xx} - g(t,x) = 0 \tag{10}$$

with one scalar sought-for parameter $\lambda \in \mathbb{R}$ and given source function $g(t,x)$. We generate observations from the exact solution and apply the above procedure to identify λ. We also investigate the performance of the parameter estimation when the observations are noisy.

2. As a more difficult example, we consider viscous Burgers' equation

$$u_t + \lambda_1 u u_x - \lambda_2 u_{xx} = 0, \quad x \in [-1;1], t \in [0,1] \tag{11}$$

with sought-for parameter $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)^{\mathrm{T}}$. This equation is nonlinear and serves as a prototype of the governing equations of fluid dynamics. As in Example 1, we investigate the performance of the parameter estimation for both clean and noisy observations.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning.* MIT press, 2016.

[2] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, 2017.

[3] D. Sivia and J. Skilling. *Data analysis: a Bayesian tutorial.* OUP Oxford, 2006.

[4] L. Wasserman. *All of Statistics.* Springer New York, 2004.