Parameter estimation of partial differential equations via neural networks

Alexander Glushko, Dmitry I. Kabanov

23 June 2019

We are interested in applying neural networks to the problem of parameter estimation of partial differential equations from given observations. Precisely, 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:

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

where $u = u(x, t)$ is the solution of the equation, $\mathcal{N}(u; \boldsymbol{\lambda})$ is a nonlinear algebraic-differential operator, $\boldsymbol{\lambda}$ is a vector of parameters. Here and below, subscript t denotes differentiation with respect to time. The goal is to estimate $\boldsymbol{\lambda}$ from the observations $\boldsymbol{D}$. We follow an approach proposed in [2]. Given observations, we train a feedforward neural network [1]

$$u(x, t; \boldsymbol{\theta}) = g_L \circ g_{L-1} \circ \dots \circ g_1, \tag{2}$$

where $\boldsymbol{\theta} = (W_1, b_1, \dots, W_L, b_L)^{\mathrm{T}}$ and

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

with σ being a so-called activation function, which must be nonlinear and is applied componentwise. In this work, we plan to use $\sigma(z) = \tanh(z)$. The hyperparameters, L (the number of the layers) and the width of each layer, are to be determined later during the course of the project. Neural network $u(x, t; \boldsymbol{\theta})$ is trained along with the derived neural network

$$f(x, t; \lambda) = u_t + \mathcal{N}(u; \boldsymbol{\lambda}), \tag{3}$$

which captures information about the observations.

The training procedure defined as follows. We assume that the observed data are

$$u_i = u(x_i, t_i; \theta) + \epsilon_i, \quad i = 1, \dots, N. \tag{4}$$

Additionally, we assume that along with each datum u_i we also observe

$$f_i = f(x_i, t_i; \boldsymbol{\lambda}) + \delta_i, \quad i = 1, \ldots, N, \tag{5}$$

where

$$f(x_i, t_i; \boldsymbol{\lambda}) = u_t + \mathcal{N}(u(x_i, t_i)); \boldsymbol{\lambda}) = 0, \quad i = 1, \ldots, N. \tag{6}$$

We denote these extended observations by $\widetilde{\boldsymbol{D}}$.

By Bayes' rule the optimal values of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ are found through maximization of the posterior distribution

$$\rho(\boldsymbol{\theta}, \boldsymbol{\lambda} | \widetilde{\boldsymbol{D}}) \propto \rho(\widetilde{\boldsymbol{D}} | \boldsymbol{\theta}, \boldsymbol{\lambda}) \times \rho(\boldsymbol{\theta}, \boldsymbol{\lambda}). \tag{7}$$

We assume that all observations are independent from each other and that u_i and f_i, $i = 1, \ldots, N$, are independent as well and that the observations are Gaussian processes:

$$\rho(u_1, \ldots, u_N | \boldsymbol{\theta}, \sigma) \propto \frac{1}{\sigma^N} \exp\left(-\frac{\sum_{i=1}^N (u_i - u(x_i, t_i; \boldsymbol{\theta}))^2}{2\sigma^2} \right) \tag{8}$$

$$\rho(f_1, \ldots, f_N | \boldsymbol{\lambda}, \tau) \propto \frac{1}{\tau^N} \exp\left(-\frac{\sum_{i=1}^N f(x_i, t_i; \boldsymbol{\lambda})^2}{2\tau^2} \right), \tag{9}$$

and assigning Jeffrey's priors for σ and τ, we obtain that

$$\rho(\tilde{\boldsymbol{D}} | \boldsymbol{\theta}, \boldsymbol{\lambda}) \propto \left(\sum_{i=1}^N (u_i - u(x_i, t_i; \boldsymbol{\theta}))^2 \right)^{-N/2} \times \left(\sum_{i=1}^N (f(x_i, t_i; \boldsymbol{\lambda}))^2 \right)^{-N/2}. \tag{10}$$

Moreover, we assume improper flat prior distribution:

$$\rho(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \text{const for all } \boldsymbol{\theta} \text{ and } \boldsymbol{\lambda}. \tag{11}$$

Using the above, the maximization of the posterior distribution is equivalent to the minimization of the log-likelihood function

$$\mathscr{L} = \text{const} - \frac{N}{2} \log \sum_{i=1}^N [u_i - u(x_i, t_i; \boldsymbol{\theta})]^2 - \frac{N}{2} \log \sum_{i=1}^N \left[f(x_i, t_i; \boldsymbol{\lambda}) \right]^2, \tag{12}$$

the minimum of which yields the optimal values of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$.

Training of neural networks in [2] lacks uncertainty quantification for the found parameters. In this work we apply bootstrap procedure [3] quantify the uncertainty by providing confidence sets for $\boldsymbol{\lambda}$.

As a concrete example, we consider linear heat equation

$$u_t - \lambda u_{xx} - g(t, x) = 0 \tag{13}$$

with one scalar sought-for parameter λ and given source function $g(t,x)$.

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

$$u_t + \lambda_1 u u_x - \lambda_2 u_{xx} = 0, \quad x \in [-1;1], t \in [0,1] \tag{14}$$

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. It is known that the solutions of this equation may develop sharp gradients in finite time even for smooth initial condition.

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