GDAŃSK UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRONICS, TELECOMMUNICATIONS AND INFORMATICS

DEPARTMENT OF ROBOTICS AND DECISION SYSTEMS

Physics-Informed Neural Network Solution of the Stochastic Burgers Equation

Mariusz Domżalski, Zdzisław Kowalczuk

Abstract

The neural network as a universal approximator can be used to model solutions to differential equations. In this presentation, we consider the problem of training a neural network in order to learn a solution to a stochastic partial differential equation (SPDE). In order for the approximation determined by the neural network to be consistent with the structure of the SPDE, apart from applying the initial and boundary conditions for learning, it is necessary to extend the penalty function by two components. The first results from the physics-informed neural network (PINN) method and is related to how well the neural network function satisfies a given differential equation. The second component is related to the modeling of the stochastic component, which can be learned after taking into account the loss function dependent on the fit of moments (expected value and covariance matrix) between the simulation data and the results obtained from the neural network. We are going to illustrate the proposed method using the Burgers' SPDE.

Stochastic Burgers Equation

We use physics-informed neural network to solve the following one-dimensional nonlinear Burgers' stochastic partial differential equation (the one dimensional Burgers' SPDE):

$$\begin{aligned} du(t,x) =& \nu\partial_x^2 u(t,x)dt - u(t,x)\partial_x u(t,x)dt + \sigma dw(t,x), \\ & t \geq 0 \end{aligned}$$

Associated Non-Stochastic PDE

$$f(t,x) = \frac{du(t,x)}{dt} + u(t,x)\partial_x u(t,x) - \nu\partial_x^2 u(t,x) = 0$$

Loss function

$$\mathrm{MSE} = \mathrm{MSE}_u + \mathrm{MSE}_f + \mathrm{MSE}_\sigma$$

$$\mathrm{MSE}_u = \frac{1}{N_u}\sum_{i=1}^{N_u} \left|\hat{u}(t_u^i, x_u^i) - u^i\right|$$

$$\mathrm{MSE}_f = \frac{1}{N_f}\sum_{i=1}^{N_u} \left|\hat{f}(t_f^i, x_f^i)\right|$$

$$\mathrm{MSE}_\sigma = \frac{1}{N_\sigma}\sum_{i=1}^{N_\sigma} \|\hat{\mu}(t_\sigma^i, x_\sigma^i) - \mu(t_\sigma^i, x_\sigma^i)\| + \|\hat{\Sigma}(t_\sigma^i, x_\sigma^i) - \Sigma(t_\sigma^i, x_\sigma^i)\|$$

Conclusions

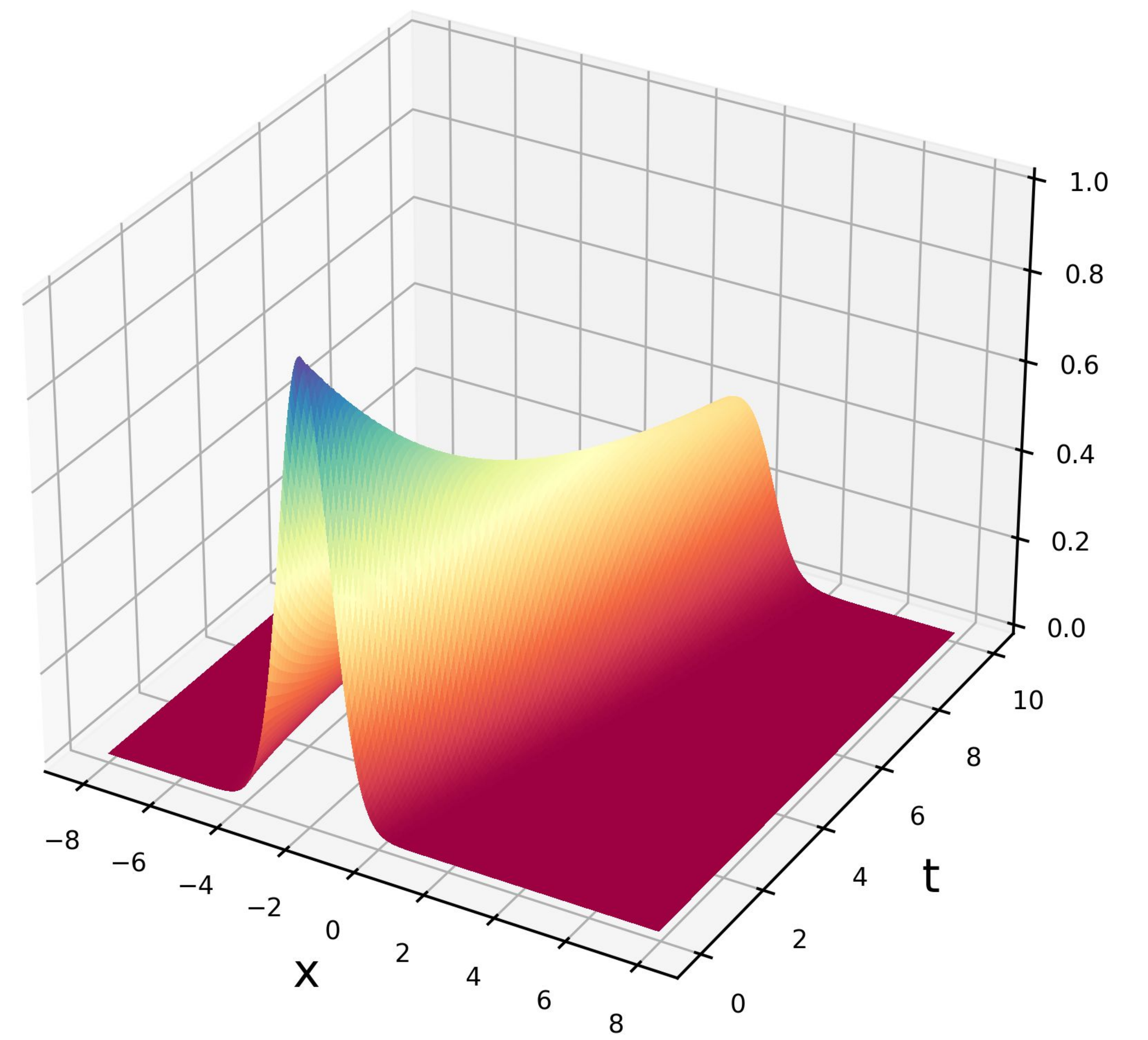
Certainly, this method, while promising, also raises a number of questions:

- what about complexity in case of complex multidimensional problems (curse of dimensionality)?
- what about convergence to the true value of the solution?
- since neural networks are sensitive to noisy data, how will this approach deal with stochastic processes?

We will try to answer these kinds of questions in the next works.

Non-Stochastic Case Example

Burgers' Equation



PINN solution