New approach for the Black–scholes equation based on Physics informed neural network

[redacted] Sunggu Kim[b], [redacted]

Abstract

In this paper, we present a new approach for Black–scholes(BS) equation based on Physics informed neural network(PINN).

Keywords: Black–Scholes equation, Physics informed neural network

1. Introduction

Machine learning and deep learning used in wide application such as image recongnition, natural language processing, and others. Among these many application, Dissanayake and Phan-Thien introduced neural network based approximations for solving partial differential equations(PDE) in [2]. This approach was recently revisited, since machine learning and deep learning have been show good performance as technology advances. Specifically, the author in [4] produced physics informed neural networks(PINN) that are trained to solve supervised learning tasks with given law of physics from PDEs.

One of PDE model is the Black–Scholes(BS) equation which first introduced by Black and Scholes [1] and Merton [3] for describe a financial market containing derivative investment instrument. As the financial market gets complex and

[redacted]

Our neural network calculate the derivatives of f by using auto differentiation as same as the PINN. Fig. 1 shows that schematic of our neural network with PDE part.

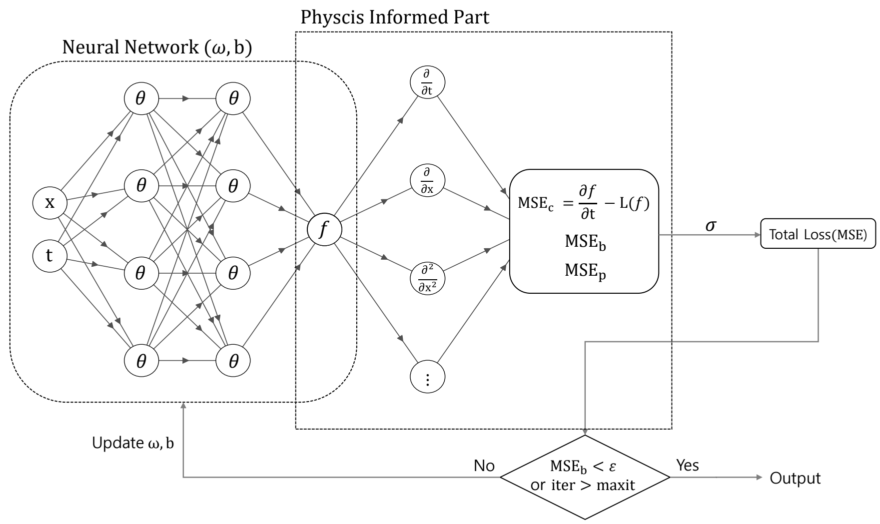


Figure 1: Schematic of our PINN along with neural network and PDE part.

3.1. Data initialization

We construct the mesh for space and time, and creating training set for our neural network. We use the computational domain as $S \in \Omega$ and $t \in D$. Here $\Omega = [0, S_{max}]$ and $D = [0, T]$, where S_{max} is the maximum price of underlying

[redacted]

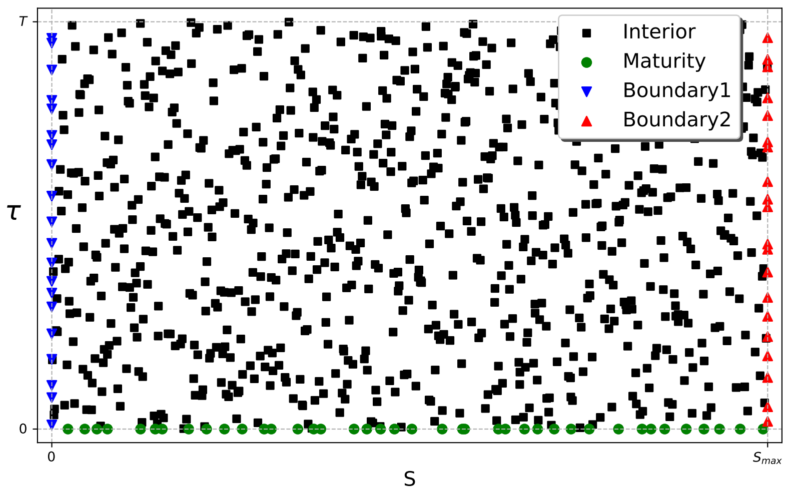


Figure 2: Selected points on computational domain. Interior points ('□') is $\{S_c^i, \tau_c^i\}$, points at maturity ('○') is $\{S_p^i\}_{i=1}^{N_p}$, and boundary points ('▽') and ('△') are $\{\tau_{b_1}^i\}_{i=1}^{N_{b_1}}$ and $\{\tau_{b_2}^i\}_{i=1}^{N_{b_2}}$, respectively.

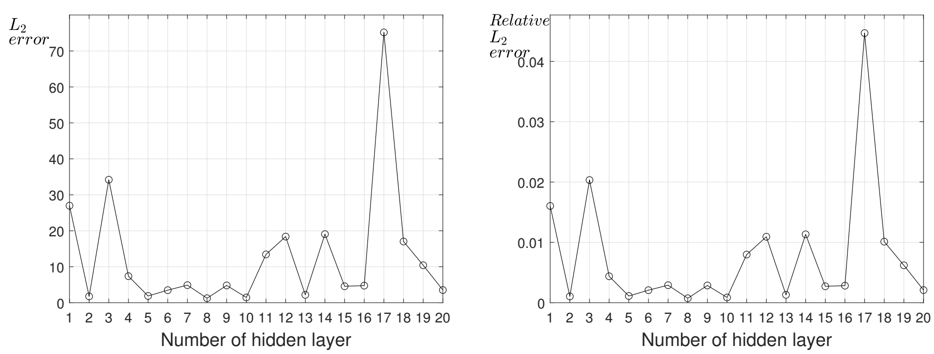
3.2. *Loss function*

[redacted]

domain, where $h = S_{max}/N_x$ is uniform grid size and N_x is the total number of space steps. Then we define the discrete L_2 error and relative L_2 error at $t = 0$ as,

$$
\begin{aligned}
||u - \hat{u}||_2 &= \sqrt{\sum_{i=1}^{N_x} (u_i - \hat{u}_i)^2 * h}, \\
||u - \hat{u}||_2 &= \sqrt{\sum_{i=1}^{N_x} (u_i - \hat{u}_i)^2 * h} \Big/ \sqrt{\sum_{i=1}^{N_x} (u_i)^2 * h}
\end{aligned}
$$

where u_i is the approximations of u at $t = 0$ and $S = x_i$.



(a) L_2 error

(b) Relative L_2 error

Figure 3: Discrete L_2 error and relative L_2 for each hidden layer number.

[redacted]