Kalpa Publications in Computing
Volume XXX, 2022, Pages 174–176
Proceedings of V XoveTIC Conference. XoveTIC 2022

Boundary-safe PINNs extension *

Joel P. Villarino[1], José A. García Rodríguez[2], and Álvaro Leitao[3]

[1] University of Coruña, A Coruña, Galicia, España
joel.perez.villarino@udc.es
[2] University of Coruña, A Coruña, Galicia, España
jose.garcia.rodriguez@udc.es
[3] Centro de Investigación CITIC, A Coruña, Galicia, España
alvaro.leitao@udc.es

Abstract

The goal of this work is to solve a nonlinear parabolic PDE problem that arise in the financial world by means of the so called PINNs methodology. We propose a novel treatment of the boundary conditions that allows us to avoid, as far as possible, the heuristic choice of the weights for the contributions of the boundary addends of the loss function that come from the boundary conditions.

1 Introduction

In recent years there has been a growing interest in approximating the solution of partial differential equations (PDEs) by means of deep neural networks (DNNs), mathematically understood as multiple chained compositions of nonlinear multivariate functions. DNNs are known for being Universal Approximators, property that has been exploited in the recent literature to solve PDEs. The DNN is trained to learn data from a physical law that is given by a PDE. As a result, a high dimensional nonlinear optimization problem is obtained. It must be solved using nonlinear optimization algorithms. Recently, with the advances in automatic differentiation algorithms (AD) and hardware (GPUs), this kind of technique has gained more momentum in the literature and, currently, the most promising approach is known as Physics-informed neural networks (PINNs), see [2]. The use of DNNs for solving PDEs has several advantages. They can be used for solving nonlinear PDEs without any extra effort; and they yield accurate approximations of the partial derivatives of the solution via AD. However, they also present certain disadvantages, such as the difficulty in the treatment of the boundary conditions.

We propose a novel approach to deal with the boundary conditions that allows us to avoid the heuristic choice of the weights for the contributions of the boundary addends to the loss function.

*Thanks to the support received from the Centro de Investigación de Galicia "CITIC", funded by Xunta de Galicia and the European Union (European Regional Development Fund- Galicia 2014-2020 Program), by grant ED431G 2019/01.

2 Problem formulation

We address a specific and challenging derivative valuation problem in the context of Counterparty Credit Risk. Let $\Omega = [0, S_1^+] \times [0, S_2^+] \subset \mathbb{R}^2$ be a bounded and connected domain and K (strike) and $T > 0$ (maturity of the contract). In addition, we define the spatial boundaries of the domain Ω as $\Gamma_1^0 = (0,T) \times \{0\} \times [0, S_2^+]$, $\Gamma_2^0 = (0,T) \times (0, S_1^+] \times \{0\}$, $\Gamma_1^+ = (0,T) \times \{S_1^+\} \times (0, S_2^+)$ and $\Gamma_2^+ = (0,T) \times (0, S_1^+] \times \{S_2^+\}$. Consider the following boundary value problem. Given a nonlinear function $f \in \mathcal{C}(\mathbb{R})$,

$$f(\hat{V}) = \lambda_B(1 - R_B)\hat{V}^- + \lambda_C(1 - R_C)\hat{V}^+ + s_F\hat{V}^+, \tag{1}$$

where λ_B, $\lambda_C \geq 0$ and $R_B, R_C \in [0,1]$ are the credit solvency parameters; and the second order elliptic operator $\mathcal{L}$,

$$\mathcal{L} = -\frac{\sigma_1^2 S_1^2}{2}\frac{\partial^2}{\partial S_1^2} - \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2}{\partial S_1 \partial S_2} - \frac{\sigma_2^2 S_2^2}{2}\frac{\partial^2}{\partial S_2^2} - r_{R_1} S_1 \frac{\partial}{\partial S_1} - r_{R_2} S_2 \frac{\partial}{\partial S_2} + r\mathcal{I}, \tag{2}$$

with $\sigma_1, \sigma_2 > 0$; $r_{R_1}, r_{R_2} \in \mathbb{R}$ and $\rho \in [-1,1]$ the Black Scholes model parameters. The goal is to find $\hat{V} : (t, S_1, S_2) \in [0,T] \times \Omega \longrightarrow \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial \hat{V}}{\partial t} + \mathcal{L}(\hat{V}) + f(\hat{V}) = 0, & \text{in } (0,T) \times \mathring{\Omega}, \\ \dfrac{\partial^2 \hat{V}}{\partial S_i^2} = 0, & \text{in } \Gamma_i^+, \ i = 1,2, \\ \hat{V} - K \exp\{-(r + \lambda_B(1 - r_B) + \lambda_C(1 - r_C))t\} = 0, & \text{in } \Gamma_i^0, \ i = 1,2, \\ \hat{V} - \max\{K - \min\{S_1, S_2\}\} = 0, & \text{in } t = 0. \end{cases} \tag{3}$$

We want to approximate the risky derivative value $\hat{V}$ under the "worst of" payoff (see the last equation). This can be done by means of a feed-forward neural network, $\hat{V}_\theta(t, S_1, S_2) := \hat{V}(t, S_1, S_2; \theta)$, where $\theta \in \mathbb{R}^P$ are the network parameters. Thus, we need to find the parameters θ that yields the best approximation. This leads to a global optimization problem that can be written in terms of the minimization of a loss function $\mathcal{J}(\theta)$,

$$\begin{aligned} \mathcal{J}(\theta) = & \frac{\lambda_{\mathcal{I}}}{S_1^+ \times S_2^+ \times T}\int_0^T \int_\Omega \left|\mathcal{R}_\theta^{\mathcal{I}}(t,x)\right|^2 dSdt + \frac{\lambda_{\Gamma_1^+}}{S_2^+ \times T}\int_0^T \int_0^{S_2^+} \left|\mathcal{R}_\theta^{\Gamma_1^+}(t, S_1^+, S_2)\right|^2 dS_2 dt \\ & + \frac{\lambda_{\Gamma_2^+}}{S_1^+ \times T}\int_0^T \int_0^{S_1^+} \left|\mathcal{R}_\theta^{\Gamma_2^+}(t, S_1, S_2^+)\right|^2 dS_1 dt + \frac{\lambda_{\Gamma_1^0}}{S_2^+ \times T}\int_0^T \int_0^{S_2^+} \left|\mathcal{R}_\theta^{\Gamma_1^0}(t, 0, S_2)\right|^2 dS_2 dt \\ & + \frac{\lambda_{\Gamma_2^0}}{S_1^+ \times T}\int_0^T \int_0^{S_1^+} \left|\mathcal{R}_\theta^{\Gamma_2^0}(t, S_1, 0)\right|^2 dS_1 dt + \frac{\lambda_{\mathcal{O}}}{S_1^+ \times S_2^+}\int_0^{S_2^+} \int_0^{S_1^+} \left|\mathcal{R}_\theta^{\mathcal{O}}(0, S_1, S_2)\right|^2 dS_1 dS_2, \end{aligned} \tag{4}$$

where $\lambda_{\mathcal{X}}$ are preset hyperparameters that allow to impose a weight to each addend of the loss; and ${\mathcal{R}_\theta}^{\mathcal{X}}$ are the left hand side terms of the boundary problem (3), which can be computed via AD. The integrals are divided by the integration volume to work with dimensionless quantities. Under such configuration, not always the optimization algorithm can get us close to a good local minima. Although the reasons why this happens are poorly understood, [1] points to the fact that training is focused on getting a small PDE residual, while having large errors in the fitting of the boundary conditions. As a solution, we propose a novel approach based on taking as a residual not the boundary condition itself, but the resulting PDE restricted to the corresponding boundary. This will produce losses of an order of magnitude similar to that produced by the interior residual. In our particular problem, our proposal translates into imposing as boundary residuals. For $i = 1, 2, \ i \neq j$,

$$\mathcal{R}_\theta^{\Gamma_i^+} = \frac{\partial \hat{V}_\theta}{\partial t} - \rho\sigma_i\sigma_j S_i S_j \frac{\partial^2 \hat{V}_\theta}{\partial S_i \partial S_j} - \frac{\sigma_j^2 S_j^2}{2}\frac{\partial^2 \hat{V}_\theta}{\partial S_j^2} - r_{R_i} S_i \frac{\partial \hat{V}_\theta}{\partial S_i} - r_{R_j} S_j \frac{\partial \hat{V}_\theta}{\partial S_j} + r\hat{V}_\theta + f(\hat{V}_\theta), \tag{5}$$

$$\mathcal{R}_{\theta}^{\Gamma_i^0} = \frac{\partial \hat{V}_\theta}{\partial t} - \frac{\sigma_j^2 S_j^2}{2} \frac{\partial^2 \hat{V}_\theta}{\partial S_j^2} - r_{R_j} S_j \frac{\partial \hat{V}_\theta}{\partial S_j} + r\hat{V}_\theta + f(\hat{V}_\theta). \tag{6}$$

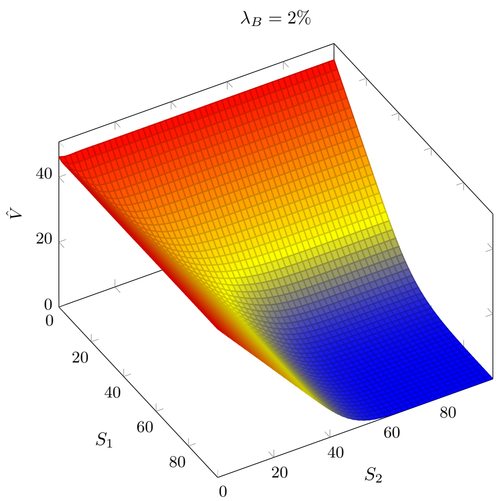
Boundary residuals given in (5) are obtained by substituting the boundary conditions given for Γ_1^+, Γ_2^+ in the PDE; while those given in (6) come from considering the PDE in Γ_1^0, Γ_2^0. This is because the Dirichlet conditions in these boundaries are the solution of the PDE on them.

3 Numerical experiment

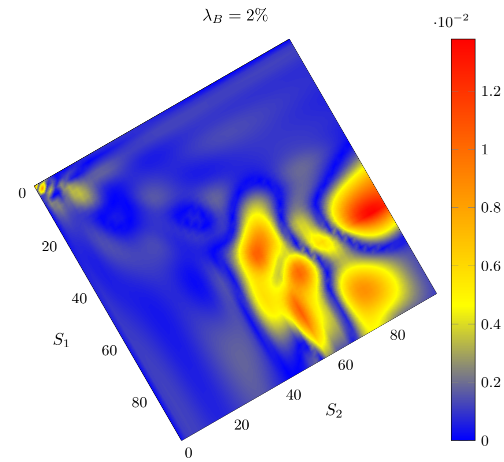
Having defined the mathematical model of (3) and how they fit under our reformulation via PINNs, we compare the computed DNN solution with a reliable reference.

We consider a 4-layers, 60-units per layer DNN and the loss function (4) with the boundary residuals defined in (5), (6). We use a grid of $141,204$ training points. The optimization process has $20,000$ steps with Adam and $2,500$ steps with L-BFGS.

In Figure 1a the PINNs solution for the problem (3) at time $t = T$ is presented, while Figure 1b shows the relative error compared with the reference values. The main source of error is given by the region in which the solution is close to zero, where a relative error of the order of 10^{-2} is observed. This is mainly due to the non-differentiability of the initial condition. In the remaining domain we have a relative error of the order of 10^{-3}.



(a) Solution surface by PINNs.



(b) Relative error colour map.

Figure 1: DNN for the problem (3) with parameters: $K = 50$, $T = 1$, $r = 0.03$, $\sigma_1 = 0.25$, $\sigma_2 = 0.15$, $r_{R_1} = 0.015$, $r_{R_2} = 0.022$, $\rho = -0.65$, $\lambda_B = 0.02$, $\lambda_C = 0.07$, $R_B = 0.5$ and $R_C = 0.3$.

References

[1] A. Karpatne, R. Kannan, and V. Kumar. *Knowledge-guided machine learning: Accelerating discovery using scientific knowledge and data.* Chapman and Hall/CRC, 1st edition, 2022.

[2] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part I): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.