Neural networks-based algorithms for option pricing

Jasper Rou

Delft University of Technology

November 4, 2022

Option pricing PDE

- Derivative $u(t, S)$ with pay-off $\Phi(S)$ at maturity T

Option pricing PDE

- Derivative $u(t, S)$ with pay-off $\Phi(S)$ at maturity T
- $u(t, S) = e^{-r(T-t)}\mathbb{E}[\Phi(S_T)|S_t = S]$

Option pricing PDE

- Derivative $u(t, S)$ with pay-off $\Phi(S)$ at maturity T
- $u(t, S) = e^{-r(T-t)}\mathbb{E}[\Phi(S_T)|S_t = S]$
- Feynman-Kac formula:

$$\frac{\partial u}{\partial t} + \mathcal{A}u = 0$$

$$u(T) = \Phi(S_T)$$

Neural network algorithm

1. Initialize θ_0

Neural network algorithm

1. Initialize θ_0
2. For each training stage:

Neural network algorithm

1. Initialize θ_0
2. For each training stage:
    1. Generate random inputs x^i

Neural network algorithm

1. Initialize θ_0
2. For each training stage:
   1. Generate random inputs x^i
   2. Calculate the cost $L(\theta_n, x^i)$

Neural network algorithm

1. Initialize θ_0
2. For each training stage:
    1. Generate random inputs x^i
    2. Calculate the cost $L(\theta_n, x^i)$
    3. Take step $\theta_{n+1} = \theta_n - \alpha_n \nabla_\theta L(\theta_n, x^i)$

Deep Galerkin Method

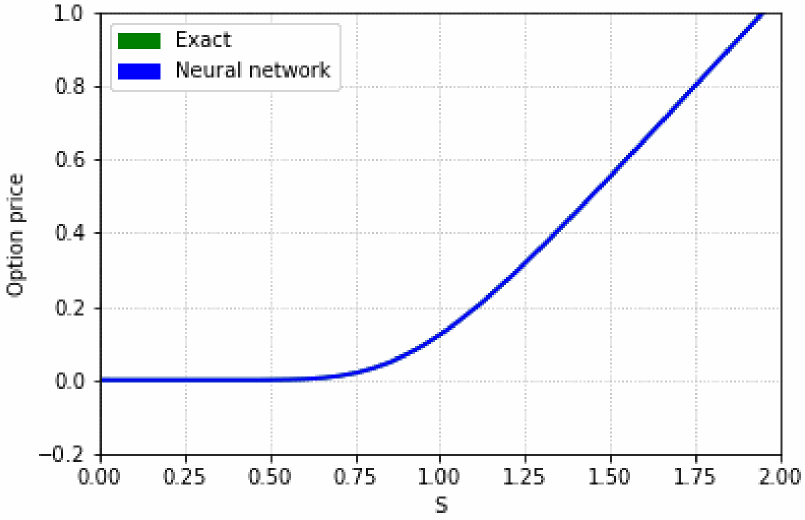
$$\frac{\partial u}{\partial t} + \mathcal{A}u = 0$$
$$u(T) = \Phi(S_T)$$

Deep Galerkin Method

$$\frac{\partial u}{\partial t} + \mathcal{A}u = 0$$
$$u(T) = \Phi(S_T)$$

$$L(\theta_n, x^i) = \left\| \frac{\partial u}{\partial t} + \mathcal{A}u \right\|^2_{L^2([0,T];\Omega)} + \| u(T) - \Phi(S_T) \|^2_{L^2(\Omega)}$$

Result



Exact
Neural network
Option price
S
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

Neural networks-based algorithms for option pricing

Jasper Rou

Delft University of Technology

November 4, 2022

j.g.rou@tudelft.nl www.jasperrou.nl