Financial option valuation by unsupervised learning with artificial neural networks

Salvador, Beatriz; Oosterlee, Cornelis W.; van der Meer, Remco

DOI
10.3390/math9010046

Publication date
2021

Document Version
Final published version

Published in
Mathematics

mathematics

MDPI

Article

Financial Option Valuation by Unsupervised Learning with Artificial Neural Networks

Beatriz Salvador [1,*], Cornelis W. Oosterlee [2,3] and Remco van der Meer [2,3]

[1] Department of Mathematics, Campus de Elviña, University of A Coruña, 15071 A Coruña, Spain
[2] CWI—Centrum Wiskunde & Informatica, 1098 Amsterdam, The Netherlands; C.W.Oosterlee@cwi.nl (C.W.O.); remcovdmeer0@gmail.com (R.v.d.M.)
[3] DIAM, Delft University of Technology, 2628 Delft, The Netherlands
* Correspondence: beatriz.salvador@udc.es

Abstract: Artificial neural networks (ANNs) have recently also been applied to solve partial differential equations (PDEs). The classical problem of pricing European and American financial options, based on the corresponding PDE formulations, is studied here. Instead of using numerical techniques based on finite element or difference methods, we address the problem using ANNs in the context of unsupervised learning. As a result, the ANN learns the option values for all possible underlying stock values at future time points, based on the minimization of a suitable loss function. For the European option, we solve the linear Black–Scholes equation, whereas for the American option we solve the linear complementarity problem formulation. Two-asset exotic option values are also computed, since ANNs enable the accurate valuation of high-dimensional options. The resulting errors of the ANN approach are assessed by comparing to the analytic option values or to numerical reference solutions (for American options, computed by finite elements). In the short note, previously published, a brief introduction to this work was given, where some ideas to price vanilla options by ANNs were presented, and only European options were addressed. In the current work, the methodology is introduced in much more detail.

Keywords: (non)linear PDEs; Black–Scholes model; artificial neural network; loss function; multi-asset options

Citation: Salvador, B.; Oosterlee, C.W.; van der Meer, R. Financial Option Valuation by Unsupervised Learning with Artificial Neural Networks. *Mathematics* **2021**, *9*, 46. https://dx.doi.org/10.3390/math9010046

Received: 13 November 2020
Accepted: 18 December 2020
Published: 28 December 2020

1. Introduction

The interest in machine learning techniques, due to the remarkable successes in different application areas, is growing exponentially. Impressive results have been achieved in image recognition or natural language processing problems, among others. The availability of large data sets and powerful compute units has brought the broad field of data science to a next level. ANNs are learning systems based on a collection of artificial neurons that constitute a connected network [1]. Such systems "learn" to perform tasks, generally without being programmed with task-specific rules. The neurons are organized in multiple layers; The input layer receives external data, the output layer produces the final result. The layers in between input and output are the so-called hidden layers [2]. Many different financial problems have also been addressed with machine learning, like stock price prediction, where ANNs are trained to detect patterns in historical data sets to predict future trends [3,4], or bond rating predictions, see [5–7].

Motivated by the universal approximation theorems [8,9], nowadays ANNs are also being used to approximate solutions to ordinary differential equations (ODEs) or partial differential equations (PDEs) [5,10–12]. Our contribution to this field consists of solving some PDEs that appear in computational finance applications with ANNs, following the unsupervised learning methodology introduced by [13] and refined in [14].

In particular, the focus will be on pricing European and American options. Different well-known methodologies have been proposed in the literature to compute these option

prices. The best known numerical methods for such convection-diffusion-type problems are the finite difference methods (FDM) and the finite element methods (FEM). A recent methodology for financial problems is found in collocation methods. An example of this methodology, used to solve the Black–Scholes equation, is based on quintic B-splines for the spatial derivative with a finite difference approximation for the time derivative [15]. In comparison with the existing well-known methods, this method has better numerical stability and lower computational costs. These methods work particularly well for low-dimensional problems, of, at most four dimensions.

A common approach to solve particularly high-dimensional option pricing problems is Monte Carlo simulation, and, for early-exercise options, the Longstaff-Schwartz regression-based Monte Carlo algorithm [16]. These simulation-based methods are based on the stochastic formulation of the pricing problem, on discretization on the corresponding stochastic differential equations (SDEs), and the approximation of (conditional) expectations. Another method to compute high-dimensional option values is given by radial basis function (RBF) approximation [17], which is a fast method with low memory requirements. With infinitely smooth RBFs the method has spectral accuracy, meaning that the required number of unknowns for a desired accuracy is relatively small. The results in [18], for American options, confirm that this scheme is promising compared to existing numerical methods. Comparing with the classical numerical methodologies, the ANN-based method is not based on a discretization of the differential equation, neither of the PDE nor the underlying SDE, and mesh generation is not required.

The financial application on which this work is focused is the valuation of financial derivatives with PDEs. Generally, we can distinguish between supervised and unsupervised machine learning techniques. Research so far has mainly focused on supervised machine learning, i.e., given input variables x and labeled output variables y, the ANN is employed to learn the mapping function from the input to the output. The goal is then to approximate the mapping function accurately, so that for new input data x', the corresponding output y' is well approximated. Such ANN methodology usually consists of two phases. During the training phase, the ANN should learn the PDE solver with input parameters and output. This (off-line) phase usually takes substantial computing time. In the testing phase, the trained model is used to very rapidly approximate solutions for other parameter sets. In [11], the authors showed that ANNs efficiently approximate the solution to the Black–Scholes equation. In [19], option values as well as the corresponding implied volatilities were directly computed with one neural network in a supervised learning approach. The authors in [10] examined whether an ANN could derive option pricing formulas based on market prices. ANN studies for American options are also found, like in [20], and in [21], where the option was formulated as a free boundary problem. In [22] the American option implied volatility and implied dividend were assessed with the help of ANNs.

The goal of the current work is to solve the financial PDEs by applying unsupervised machine learning techniques. In such a case, only the inputs of the network are known, and based on a suitable loss function that needs to be minimized, the ANN should "converge" to the solution of the PDE problem. The ANN should learn solutions that satisfy constraints that are imposed by the PDE and the boundary conditions, without using any information about the true solution. These constraints are typically formulated as soft constraints, that are satisfied by minimizing some loss function. The potential advantage of applying ANNs to address PDE problems, instead of using classical numerical methods, is found in the problem's dimensionality. An ANN-based methodology does not suffer much from the curse of dimensionality. The authors of [13,21] provide evidence that for the well-known Poisson and Burgers equations, these unsupervised learning methods yield accurate results. The authors in [14] extended the class of PDE solutions that may be approximated by these unsupervised learning methods, by translating the PDEs to a suitably weighted minimization problem for the ANNs to solve. Moreover, in [8,9] American options were formulated as optimal stopping problems, where optimal stopping decisions were learned

and so-called ANN regression was used to estimate the continuation values. This is an example of the unsupervised learning approach to solve a specific formulation of options with early-exercise features.

European and American options modeled by the Black–Scholes PDE will be priced and solutions for all future time points and stock values will be determined. So, linear and nonlinear partial differential equations need to be solved. We will solve European and American option problems based on one and two underlying assets, as the methodology is easily extended to solving multi-asset options. For the European problems, the accuracy of the network can be measured as the analytic Black–Scholes solution will serve as a reference. American options will be formulated as linear complementarity problems. Since an analytic solution is not known in this case, the reference solutions are obtained by finite element computations on fine meshes.

This paper is organized as follows. In Section 2, the methodology to train the neural network is introduced. Moreover, the financial PDE problems are formulated, for the linear and the nonlinear case. Numerical results, ANN convergence and solution accuracy, are presented in Section 3. Finally, Section 4 concludes.

2. Methodology

In this section, in the first part, the methodology for solving linear and nonlinear time-dependent PDEs by means of unsupervised learning with artificial neural networks is introduced. Then, in the second subsection, the pricing models for European and American options are presented and the methodology to solve the PDEs by the ANNs introduced will be applied.

2.1. Artificial Neural Networks Solving PDEs

Here, the methodology is introduced, following [14], to solve linear and nonlinear time-dependent PDEs by ANNs. With this aim, the general PDE problem can be written, as follows:

$$\begin{aligned} \mathcal{N}_I(v(t,x)) &= 0, \quad x \in \widetilde{\Omega},\ t \in [0,T], \\ \mathcal{N}_B(v(t,x)) &= 0, \quad x \in \partial\widetilde{\Omega},\ t \in [0,T], \\ \mathcal{N}_0(v(t^*,x)) &= 0, \quad x \in \widetilde{\Omega} \text{ and } t^* = 0 \text{ or } t^* = T, \end{aligned} \tag{1}$$

where $v(t,x)$ denotes the solution of the PDE, $\mathcal{N}_I(\cdot)$ is a linear or nonlinear time-dependent differential operator, $\mathcal{N}_B(\cdot)$ is a boundary operator, $\mathcal{N}_0(\cdot)$ is an initial or final time operator, $\widetilde{\Omega}$ is a subset of $\mathbb{R}^D$ and $\partial\widetilde{\Omega}$ denotes the boundary on the domain $\widetilde{\Omega}$.

As mentioned in the introduction, European and American option values will be computed for one and two underlying assets by unsupervised learning. The goal is to obtain $\hat{v}(t,x)$ by minimizing a suitable loss function $L(v)$ over the space of k-times differentiable functions, where k depends on the order of the derivatives in the PDE, i.e.,

$$\arg\min_{v \in \mathcal{C}^k} L(v) = \hat{v}\,, \tag{2}$$

where $\hat{v}(t,x)$ denotes the true solution of the PDE.

Results are available that establish a relation between the value of the loss function and the accuracy of the approximated solution. The solution of the PDEs is approximated with a deep neural network. The deep neural network consists of an input layer with d neurons and the output layer consists of a single neuron, representing the entire solution of the PDE. The ANN should approximate the solution, satisfying the restrictions imposed by the PDE and the boundary conditions. A general expression for the loss function, defined in terms of the L^p norm, including a weighting, is defined as follows [13,14]:

$$L(v) = \lambda \int_{\Omega} \mid \mathcal{N}_I(v(t,x)) \mid^p dxdt + (1-\lambda) \int_{\partial\Omega} (\mid \mathcal{N}_B(v(t,x)) \mid^p + \mid \mathcal{N}_0(v(t,x)) \mid^p) dxdt, \tag{3}$$

where $\Omega = \widetilde{\Omega} \times [0,T]$, $\partial\Omega$ the boundary of Ω and

$$\begin{aligned} &\mathcal{N}_I(v(t,x)) \equiv N(v(t,x)) - F(t,x) \quad \text{in } \Omega\,, \\ &\mathcal{N}_B(v(t,x)) \equiv B(v(t,x)) - G(t,x) \quad \text{on } \partial\widetilde{\Omega}\,, \\ &\mathcal{N}_0(v(t^*,x)) \equiv H(x) - v(t^*,x) \quad \text{in } \widetilde{\Omega} \times t^*, \text{ with } t^* = 0 \text{ or } t^* = T. \end{aligned} \tag{4}$$

The integrals of the loss function are labeled as:

$$L_I(v) \equiv \int_{\Omega} \mid \mathcal{N}_I(v(t,x)) \mid^p dxdt, \tag{5}$$

and

$$L_B(v) \equiv \int_{\partial\Omega} (\mid \mathcal{N}_B(v(t,x)) \mid^p + \mid \mathcal{N}_0(v(t,x)) \mid^p) dxdt, \tag{6}$$

which are denoted as the interior and the boundary loss functions, respectively.

Financial options with early-exercise features give rise to free boundary PDE problems. Free boundary problems are well-known and often appearing in a variety of engineering problems. Next, some classical formulations of free boundary problems are recalled:

- an optimal stopping time problem,
- a linear complementarity problem (LCP),
- a parabolic variational inequality,
- a penalty problem.

Our goal here is the reformulation of the free boundary problem as a LCP, with the aim to solve this formulation by ANNs and unsupervised learning. The generic LCP formulation reads,

$$\begin{aligned} &\mathcal{N}_I(v(t,x)) \leq 0, \quad x \in \widetilde{\Omega},\, t \in [0,T], \\ &\mathcal{N}_0(v(t,x)) \geq 0, \quad x \in \widetilde{\Omega},\, t \in [0,T], \\ &\mathcal{N}_I(v(t,x)) \cdot \mathcal{N}_0(v(t,x)) = 0, \quad x \in \widetilde{\Omega},\, t \in [0,T], \\ &\mathcal{N}_B(v(t,x)) = 0, \quad \text{on } \partial\widetilde{\Omega}, \\ &\mathcal{N}_0(v(t^*,x)) = 0, \quad x \in \widetilde{\Omega} \text{ and } t^* = 0 \text{ or } t^* = T. \end{aligned} \tag{7}$$

or, equivalently,

$$\begin{aligned} \max(\mathcal{N}_0(v(t,x)), \mathcal{N}_I(v(t,x))) = 0, \quad & x \in \widetilde{\Omega},\, t \in [0,T], \\ \mathcal{N}_B(v(t,x)) = 0, \quad & \text{on } \partial\widetilde{\Omega}, \\ \mathcal{N}_0(v(t^*,x)) = 0, \quad & x \in \widetilde{\Omega} \text{ and } t^* = 0 \text{ or } t^* = T. \end{aligned} \tag{8}$$

Our expression for the loss function, to solve the linear complementarity problem, is as follows:

$$L(v) = \lambda \int_{\Omega} \mid \max(\mathcal{N}_0(v(t,x)), \mathcal{N}_I(v(t,x))) \mid^p dxdt + (1-\lambda) \int_{\partial\Omega} (\mid \mathcal{N}_B(v(t,x)) \mid^p + \mid \mathcal{N}_0(v(t,x)) \mid^p) dxdt\,. \tag{9}$$

As an alternative loss function for the LCP, a variance normalization loss function has also been considered [14], which is defined as:

$$\begin{aligned} L(v) &= \frac{\int_\Omega \mid \max(\mathcal{N}_0(v(t,x)), \mathcal{N}_I(v(t,x))) \mid^p dx}{\int_\Omega (\max(\mid \mathcal{N}_0(v(t,x)) \mid, \hat{\mathcal{N}}_I(v(t,x))))^p dx} \\ &+ \frac{\int_{\partial\Omega} (\mid \mathcal{N}_B(v(t,x)) \mid^p + \mid \mathcal{N}_0(v(t,x)) \mid^p) dxdt}{\int_{\partial\Omega} \mid v(t,x) - \bar{v} \mid^p dxdt}, \end{aligned} \tag{10}$$

where $\hat{\mathcal{N}}_I$ is defined as $\mathcal{N}_I$ but considering each term in absolute value and $\bar{v}$ is the mean of v over the corresponding domain.

The parameter $\lambda \in (0,1)$ in the loss functions represents the relative importance of the interior and boundary functions in the minimization process. The choice of such value can be addressed in different ways, see [13,14]. In this work, the loss weight is, in most of the tests, set equal to $\lambda = 0.5$. It was found in [14] that this choice works very well for PDE problems with smooth, non-oscillatory solutions (as we also encounter them in the option valuation problems under consideration). The results with the basic loss function formulation will be compared to the variance normalization loss function, for some linear complementarity problems. In addition, for some other cases, a loss function based on so-called optimal loss weights (as in [14]) will be considered.

Based on the loss function, the ANN has been trained with the Broyden-Fletcher-Goldfarb-Shanno optimization (BFGS). This is a quasi-Newton method which employs an approximate Hessian matrix. Particularly, the L-BFGS algorithm is used to optimize the vector θ, which contains all parameters defining the neural network. The activation function used in the ANN is the hyperbolic tangent function *tanh(x)*, however, other choices of the activation function can also be used, like the sigmoid function (resulting in very similar results in this work). In this work, relatively small neural networks are considered, which are formed by four hidden layers with 20 neurons each for the European and American options. Increasing the number of layers did not improve the accuracy of the solution significantly for these particular problems. Finally, the integral terms in the loss function are approximated by Monte Carlo techniques.

2.2. Financial Derivative Pricing Partial Differential Equations

Below, the option pricing partial differential equation problems are presented and the models are briefly introduced.

2.2.1. European Options, One Underlying Asset

The reference option pricing PDE for the valuation of a plain vanilla European, put or call, option is the Black–Scholes equation. The underlying asset S_t is assumed to pay a constant dividend yield δ, and follows the geometric Brownian motion:

$$dS_t = (\mu - \delta) S_t dt + \sigma S_t dW_t^P, \tag{11}$$

where W_t^P is a Brownian motion. The drift term μ, the risk-free interest rate, r, and the asset volatility, σ, are known functions. Assuming there are no arbitrage opportunities, the European option value follows from the Black–Scholes equation,

$$\begin{cases} \mathcal{L}(v) = \partial_t v + \mathcal{A}v - rv = 0, & S \in \widetilde{\Omega}, t \in [0,T), \\ v(T,S) = H(S), \end{cases} \tag{12}$$

where operator $\mathcal{A}$ is defined as,

$$\mathcal{A}v \equiv \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 v}{\partial S^2} + (r - \delta) S \frac{\partial v}{\partial S} \tag{13}$$

and function H denotes the option's payoff, which is given by:

$$\begin{cases} (K-S)^+ & \text{for a put option} \\ (S-K)^+ & \text{for a call option}\,, \end{cases} \tag{14}$$

with K the strike price in the option contract.

In order to apply numerical methods to solve the PDE, a bounded domain should be considered and a proper set of boundary conditions should be imposed. A large enough computational domain is considered, being $[0, S_\infty]$, with S_∞ four times the strike K. Depending on the kind of option, call v_c or put v_p, the problem (12) is subject to the conditions:

$$\begin{cases} v_c(t,0) = 0 \\ v_c(t,S_{\max}) = S_{\max} - Ke^{-r(T-t)}\,, \end{cases} \qquad \begin{cases} v_p(t,0) = Ke^{-r(T-t)} \\ v_p(t,S_{\max}) = 0\,. \end{cases} \tag{15}$$

The analytic solution for (12) is known:

$$v_c(t,S) = S\exp(-\delta(T-t))N_{0,1}(d_1) - K\exp(-r(T-t))N_{0,1}(d_2), \tag{16}$$

$$v_p(t,S) = K\exp(-r(T-t))N_{0,1}(-d_2) - S\exp(-\delta(T-t))N_{0,1}(-d_1), \tag{17}$$

with,

$$d_1 = \frac{\log(S/K) + (r - \delta + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}\,,\; d_2 = \frac{\log(S/K) + (r - \delta - \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}} \tag{18}$$

and $N_{0,1}(x)$ the distribution function of a standard $\mathcal{N}(0,1)$ random variable. Regarding the numerical solution with ANNs, the methodology introduced in the previous section will be employed. The operator $\mathcal{N}_I(\cdot)$ corresponds to the operator $\mathcal{L}(\cdot)$, the operator defining the boundary conditions, $\mathcal{N}_B(\cdot)$, can be deduced from (15) and is equal to $v(t,x) - G(t,x)$. Similarly, the operator for the initial condition, $\mathcal{N}_B(\cdot)$, is given by $v(t,x) - H(x)$. In particular, the loss function is defined as:

$$\begin{aligned} L(v) = \lambda \int_\Omega \mid \mathcal{L}(v(t,x)) \mid^p dxdt \\ + (1-\lambda)\int_{\partial\Omega}(\mid v(t,x) - G(t,x) \mid^p + \mid v(t,x) - H(x) \mid^p)dxdt, \end{aligned} \tag{19}$$

where functions G and H denote the values of the spatial boundary conditions and final condition, respectively. The integral terms in the loss function are approximated by Monte Carlo techniques, as a result, we obtain the following interior and boundary loss function for the parameter vector θ:

$$\begin{aligned} \widehat{L}(\theta) =& \lambda\frac{1}{n_I}\sum_{i=1}^{n_I} \mid \mathcal{L}(v(\mathbf{y}_i^I,\theta))) \mid^p + \\ & (1-\lambda)\left(\frac{1}{n_B}\sum_{i=1}^{n_B} \mid v(\mathbf{y}_i^B,\theta) - G(\mathbf{y}_i^B) \mid^p + \frac{1}{n_0}\sum_{i=1}^{n_0} \mid v(\mathbf{y}_i^0,\theta) - H(\mathbf{x}_i^0) \mid^p\right). \end{aligned} \tag{20}$$

The collocation points $\{\mathbf{y}_i^I\}_{i=1}^{n_I}$ and $\{\mathbf{y}_i^B\}_{i=1}^{n_B}$ are uniformly distributed over the domain Ω and the boundary $\partial\widetilde{\Omega}$ and $\{\mathbf{y}_i^0\}_{i=1}^{n_0}$ are uniformly distributed over the domain $T \times \widetilde{\Omega}$, respectively, and $\mathbf{y} = (t,x)$.

2.2.2. European Options, Two Underlying Assets

The model for one underlying asset is extended to valuing basket options with two underlying assets. The two-asset prices follow the following dynamics,

$$dS_{1_t} = (\mu_1 - \delta_1)S_{1_t}dt + \sigma_1 S_{1_t}dW_t^1, \tag{21}$$

$$dS_{2_t} = (\mu_2 - \delta_2)S_{2_t}dt + \sigma_2 S_{2_t}dW_t^2, \tag{22}$$

where μ_1, μ_2 are drift terms, δ_1, δ_2 dividend yields, the Brownian increments, dW^i for $i = 1, 2$, satisfy $\mathbb{E}(dW^i) = 0$, and the underlying assets are correlated:

$$\text{corr}(W^1, W^2) = \rho t \qquad \text{or} \qquad \mathbb{E}(dW^1, dW^2) = \rho dt\,. \tag{23}$$

In the Black–Scholes framework, the two-asset European option price, $v(t, S_1, S_2)$, satisfies the following PDE:

$$\begin{cases} \mathcal{L}_2(v) = \partial_t v + \mathcal{B}v - rv = 0 \quad (S_1, S_2) \in \widetilde{\Omega}\,, \quad t \in [0, T), \\ v(T, S_1, S_2) = H_2(S_1, S_2)\,, \end{cases} \tag{24}$$

where the operator $\mathcal{B}$ is defined as follows:

$$\begin{aligned} \mathcal{B}v \;\equiv\; & \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 v}{\partial S_1^2} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 v}{\partial S_2^2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 v}{\partial S_1 \partial S_2} \\ + \;& (r - \delta_1)S_1 \frac{\partial v}{\partial S_1} + (r - \delta_2)S_2 \frac{\partial v}{\partial S_2}, \end{aligned} \tag{25}$$

and function $H_2(S_1, S_2)$ denotes the payoff function. By prescribing different payoff functions, different options can be defined, like an exchange option, rainbow option or an average put option. In this work, the exchange option is addressed, for which an analytic solution is given by the Margrabe's formula [23] and the max-on-call rainbow option, for which a closed-form expression was introduced in [24,25]. These particular options are defined by their payoff functions:

$$H_2(S_1, S_2) = (S_1 - S_2)^+ \text{ exchange option,} \tag{26}$$

$$H_2(S_1, S_2) = (\max(S_1, S_2) - K)^+ \text{ max-on-call rainbow option.} \tag{27}$$

According to the Margrabe's formula, the fair value of a European exchange option at time t is given by:

$$v(t, S_1, S_2) = e^{-\delta_1(T-t)}S_1(t)N_{0,1}(d_1) - e^{-\delta_2(T-t)}S_2(t)N_{0,1}(d_2) \tag{28}$$

where $N_{0,1}$ again denotes the cumulative distribution function for the standard normal,

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho} \tag{29}$$

and

$$d_1 = (\log(S_1(t)/S_2(t)) + (\delta_2 - \delta_1 + \sigma^2/2)(T - t)/\sigma\sqrt{T - t} \quad , \quad d_2 = d_1 - \sigma\sqrt{T - t}\,. \tag{30}$$

With the following parameters:

$$d_i = \frac{\log(S_i/k) + (r - \delta_i + \frac{\sigma_i^2}{2})(T - t)}{\sigma_i\sqrt{T - t}}\,, \tag{31}$$

$$\rho_1 = \frac{\sigma_1 - \rho\sigma_2}{\sigma} \quad \text{and} \quad \rho_2 = \frac{\sigma_2 - \rho\sigma_1}{\sigma}\,, \quad i = 1, 2, \tag{32}$$

the closed-form formula for a call on the maximum is given by:

$$v_c^{\max}(t,S_1,S_2) = S_1 e^{-\delta_1(T-t)} M(d_1,d;\rho_1) + S_2 e^{-\delta_2(T-t)} M(d_2,-d+\sigma\sqrt{T-t};\rho_2) - Ke^{-r(T-t)}(1-M(-d_1+\sigma_1\sqrt{T-t},-d_2+\sigma_2\sqrt{T-t};\rho)), \quad (33)$$

where M is the cumulative bivariate normal distribution

$$M(a,b;\rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\infty}^{a}\int_{-\infty}^{b} e^{-\frac{x^2-2\rho xy+y^2}{2(1-\rho)}} dxdy\,. \quad (34)$$

To obtain a numerical solution of the PDE (24), the domain is bounded and appropriate boundary conditions are imposed. The computational domain should be sufficiently large, $[0,S_{1\infty}]\times[0,S_{2\infty}]$, where $S_{1\infty}=S_{2\infty}=4K$ (K the option strike). In the particular case of the exchange and rainbow max-on-call options, where the analytic solutions are known, the analytic option value is imposed as boundary conditions on each boundary.

As in the one-dimensional problem, the European exchange option problem is addressed building the loss function as a sum of the interior and boundary loss functions, using $\lambda = 0.5$.

2.2.3. American Options, One Underlying Asset

As introduced in Section 2.1, the problem for an American option depending on one underlying asset price is studied, particularly by means of the linear complementarity formulation.

Here, the linear complementarity problem (LCP) American option valuation formulation will be considered, see, for example, [26,27], as follows,

$$\begin{cases} \mathcal{L}(v) = \partial_t v + \mathcal{A}v - rv \le 0\,, & S\in\widetilde{\Omega}\,, t\in[0,T)\,,\\ v(t,S)\ge H(S),\\ \mathcal{L}(v)(v-H)=0,\\ v(T,S)=H(S)\,. \end{cases} \quad (35)$$

This LCP can be rewritten as a nonlinear PDE as follows

$$\begin{cases} \max\{H(S)-v(t,S),\mathcal{L}(v)\}=0\,, & S\in\widetilde{\Omega}\,, t\in[0,T)\,,\\ v(T,S)=H(S)\,. \end{cases} \quad (36)$$

Essentially, using the same methodology for solving the European option PDEs, we address the linear complementarity formulation and its equivalent formulation as a nonlinear PDE given by (36).

Following Section 2.1, the loss function can be formulated using variance normalization. Moreover, in case of the American option, λ will be also computed as the optimal loss weight.

Similar to the European case, the operators that define the linear complementarity problem in (7) can be directly identified. In particular, $\mathcal{N}_I(\cdot)=\mathcal{L}(\cdot)$, the operator equation defining the boundary condition, $\mathcal{N}_B(v(t,x))=v(t,x)-G(t,x)$, and the operator equation for the initial condition is given by $\mathcal{N}_B(v(t,x))=v(t,x)-H(x)$.

The loss function based on variance normalization depends on the variance of the network output. For the Black–Scholes American option problem, with the expressions for the operators involved in (10), the loss function following variance normalization is given by

$$\begin{aligned} L(v) &= \frac{\int_\Omega \mid \max(H(x)-v(t,x),\mathcal{L}(v(t,x))) \mid^p dx}{\int_\Omega(\max(\mid H(x)-v(t,x) \mid,\tilde{\mathcal{L}}(v(t,x))))^p dx} \\ &+ \frac{\int_{\partial\Omega}(\mid v(t,x)-G(t,x) \mid^p + \mid v(t,x)-H(x) \mid^p)dxdt}{\int_{\partial\Omega} \mid v(t,x)-\bar{v} \mid^p dxdt}, \end{aligned} \tag{37}$$

where $\tilde{\mathcal{L}}(v(t,x))$ is defined as follows

$$\tilde{\mathcal{L}}(\hat{v}) = \mid \partial_t\hat{v} \mid + \mid \frac{1}{2}\sigma^2S^2\partial^2_{SS}\hat{v} \mid + \mid (r-\delta)S\partial_S\hat{v} \mid + \mid r\hat{v} \mid, \tag{38}$$

function G refers to the boundary conditions imposed in a bounded domain which are defined as in (15) and function H denotes the final condition. Moreover, $\bar{v}$ is the mean of v over the corresponding domain, which is given as

$$\bar{v} = \frac{1}{\|\partial\Omega\|}\int_{\partial\Omega} v(t,x)dxdt\,. \tag{39}$$

Then, approximating each integral term by Monte Carlo techniques the resulting function is defined as follows

$$\begin{aligned} \widehat{L}(\theta) =& \frac{\sum_{i=1}^{n_I} \mid \max(H(x_i^I)-v(\mathbf{y}_i^I,\theta),\mathcal{L}(v(\mathbf{y}_i^I,\theta))) \mid^p}{\sum_{i=1}^{n_I}\max(\mid H(x_i^I)-v(\mathbf{y}_i^I,\theta) \mid,\tilde{\mathcal{L}}(v(\mathbf{y}_i^I,\theta)))^p} + \\ & \frac{\frac{1}{n_B}\sum_{i=1}^{n_B} \mid v(\mathbf{y}_i^B,\theta)-G(\mathbf{y}_i^B) \mid^p + \frac{1}{n_0}\sum_{i=1}^{n_0} \mid v(\mathbf{y}_i^0,\theta)-H(\mathbf{x}_i^0) \mid^p}{\frac{1}{n_*}\sum_{i=1}^{n_*} \mid v(\mathbf{y}_i^*,\theta)-\frac{1}{n_*}\sum_{j=1}^{n_*}v(\mathbf{y}_j^*) \mid^p}, \end{aligned} \tag{40}$$

with θ containing all parameters of the neural network, vector $\mathbf{y} = (t,x)$, and the collocation points $\{\mathbf{y}_i^*\}_{i=1}^{n_*}$ are uniformly distributed over the boundary $\partial\Omega$.

An alternative is to build the loss function based on an optimal loss weight. However, optimizing λ can be nontrivial.

In order to find the optimal loss weight, a so-called ϵ-close solution to the true solution $\hat{v}$ is aimed at, see [14],

$$\left|\frac{\partial^n v}{\partial y_i^n} - \frac{\partial^n\hat{v}}{\partial y_i^n}\right| \leq \epsilon\frac{\partial^n\hat{v}}{\partial y_i^n}, \tag{41}$$

for all $n \geq 0$ and $i \in 1,\ldots,d$, where d is the dimension of the problem. Satisfying such condition, the value of the optimal loss weight λ^* should be:

$$\lambda^* = \frac{\int_{\partial\Omega} \mid \hat{v}(t,x) \mid^p d\gamma}{\int_\Omega(\hat{\mathcal{N}}_I(t,x,\hat{v}))^p dxdt + \int_{\partial\Omega} \mid \hat{v}(t,x) \mid^p dxdt}, \tag{42}$$

where function $\hat{\mathcal{N}}_I(x,\hat{v})$ is defined as the function $\mathcal{N}_I(x,\hat{v})$ with each term in absolute value. This expression of λ^* is constant when the analytical solution is known. However, for the American options where the analytical solution is not known, the optimal loss weight can be computed by approximating the value of $\hat{v}$ in (42) by the trained solution. Note that in this case, the loss weight is a function instead of a constant value and is optimized by the neural network.

As a result, the loss function is built in the following way:

$$\begin{aligned} L(v) =& \lambda^*L_I(v) + (1-\lambda^*)L_B(v) \\ =& \lambda^*\int_\Omega \mid \max(H(x)-v(t,x),\mathcal{L}(v(t,x))) \mid^p dxdt + \\ & (1-\lambda^*)\int_{\partial\Omega}(\mid v(t,x)-G(t,x) \mid^p + \mid v(t,x)-H(x) \mid^p)dxdt, \end{aligned} \tag{43}$$

and the optimal loss weight is given in terms of the trained solution v, as follows:

$$\lambda^* = \frac{\int_{\partial\Omega} \mid v(t,x) \mid^p d\gamma}{\int_\Omega (\tilde{\mathcal{L}}_1(v))^p d\Omega + \int_{\partial\Omega} \mid v(t,x) \mid^p d\gamma}, \tag{44}$$

with

$$\tilde{\mathcal{L}}_1(v) = \max(\mid H(x) - v(t,x) \mid, \tilde{\mathcal{L}}(v)), \tag{45}$$

where $\tilde{\mathcal{L}}(v)$ is defined as in (38).

Comparing the loss functions for the European and American options, notice that for the American options, the loss functions are defined in terms of the nonlinear operator, which is more complicated to minimize.

2.2.4. Two-Asset American Option

The one underlying asset American option pricing problem is extended to also price multi-asset American options. The focus is, on two underlying assets, again formulating the problem as a linear complementarity problem. Based on two asset prices following correlated geometric Brownian motion, the American option value can be modeled by the following linear complementarity problem:

$$\begin{cases} \mathcal{L}_2(v) = \partial_t v + \mathcal{B}v - rv \leq 0\,, & (S_1, S_2) \in \widetilde{\Omega}\,, t \in [0,T), \\ v(t,S_1,S_2) \geq H_2(S_1,S_2), \\ \mathcal{L}_2(v)(v - H_2) = 0, \\ v(T,S_1,S_2) = H_2(S_1,S_2)\,. \end{cases} \tag{46}$$

Operator $\mathcal{B}$ is defined as in (25) and function $H_2(S_1,S_2)$ denotes the payoff function. In order to compare with the European option problem, an American call on the maximum is also priced, moreover, a two-asset spread option and a put arithmetic average option are addressed. The corresponding payoff functions are defined as:

$$H_2(S_1,S_2) = (\max(S_1,S_2) - K)^+, \text{ max-on-call rainbow option,} \tag{47}$$

$$H_2(S_1,S_2) = (S_1 - S_2 - K)^+, \text{ asset spread option,} \tag{48}$$

$$H_2(S_1,S_2) = (K - (S_1+S_2)/2)^+, \text{ arithmetic average put.} \tag{49}$$

In order to solve the linear complementarity formulation by numerical methods, a bounded domain should be considered and appropriate boundary conditions should be imposed. In particular, a sufficiently large domain will be considered to avoid that the solution is affected by the conditions, in the interested regions of the asset prices. Whereas for the European option problem the analytical solution is known and imposed as a boundary condition, for the American options problem, where the analytical solution is not known, the appropriate boundary conditions should be defined. Following [28], based on the theory of Fichera [29], the following notation is introduced, $x_0 = \tau$, $x_1 = S_1$, $x_2 = S_2$, and the domain $\Omega^* = (0, x_0^\infty) \times (0, x_1^\infty) \times (0, x_2^\infty)$, where $x_0^\infty = T$, $x_1^\infty = S_{1\infty}$ and $x_2^\infty = S_{2\infty}$. The boundary of Ω^* is,

$$\partial\Omega^* = \bigcup_{i=0}^{2} (\Gamma_i^{*,-} \cup \Gamma_i^{*,+}), \tag{50}$$

with the notation:

$$\Gamma_i^{*,-} = \{(x_0,x_1,x_2) \in \partial\Omega^*\,, x_i = 0\}, \tag{51}$$

$$\Gamma_i^{*,+} = \{(x_0,x_1,x_2) \in \partial\Omega^*\,, x_i = x_i^\infty\}\,. \tag{52}$$

Then, the PDE in (46) can be written in the form:

$$\sum_{i,j=0}^{2} b_{i,j}\frac{\partial^2 v}{\partial x_i \partial x_j} + \sum_{j=0}^{2} p_j \frac{\partial v}{\partial x_j} + c_0 v \leq g_0, \tag{53}$$

where the involved data are defined as follows:

$$B(x_0, x_1, x_2) = (b_{ij}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2}\sigma_1^2 x_1^2 & \frac{\rho\sigma_1\sigma_2 x_1 x_2}{2} \\ 0 & \frac{\rho\sigma_1\sigma_2 x_1 x_2}{2} & \frac{1}{2}\sigma_2^2 x_2^2 \end{pmatrix}, \qquad c_0 = r\,, \tag{54}$$

$$p(x_0, x_1, x_2) = (p_j) = \begin{pmatrix} -1 \\ (r-\delta_1)x_1 \\ (r-\delta_2)x_2 \end{pmatrix}, \qquad g(x_0, x_1, x_2) = 0\,. \tag{55}$$

Next, the following subset of Γ^* is introduced in terms of the normal vector to the boundary pointing inwards Ω^*, $\overrightarrow{m} = (m_0, m_1, m_2)$

$$\Sigma^0 = \left\{ x \in \partial\Omega^* / \sum_{i,j=0}^{2} b_{ij} m_i m_j = 0 \right\}, \quad \Sigma^1 = \partial\Omega^0 - \Sigma^0\,, \tag{56}$$

$$\Sigma^2 = \left\{ x \in \Sigma^0 / \sum_{i=0}^{2} \left(p_i - \sum_{j=0}^{2} \frac{\partial b_{ij}}{\partial x_j} \right) m_i \leq 0 \right\}. \tag{57}$$

In this particular problem, the subsets are: $\Sigma^0 = \Gamma_0^{*,-} \cup \Gamma_0^{*,+} \cup \Gamma_1^{*,-} \cup \Gamma_2^{*,-}$, $\Sigma^1 = \Gamma_1^{*,+} \cup \Gamma_2^{*,+}$ and $\Sigma^2 = \Gamma_0^{*,+}$. Thus, following [28], the boundary conditions must be imposed over the subset $\Sigma^1 \cup \Sigma^2$ which matches with the set $\Gamma_0^{*,-} \cup \Gamma_1^{*,+} \cup \Gamma_2^{*,+}$. Then, it is not necessary to impose boundary conditions above the boundary where the asset prices S_1 and S_2 are equal to zero. Moreover, for simplicity, the option value is assumed equal to the payoff when the asset prices S_1 and S_2 take the maximum values.

Next, taking into account the methodologies proposed to solve the one-dimensional American problem and the two-dimensional European problem, the loss function to solve the multi-asset American option by artificial neural network is defined. First of all, the linear complementarity problem (46) is rewritten as the equivalent nonlinear PDE:

$$\begin{cases} \max\{H_2(S_1, S_2) - v(t, S_1, S_2), \mathcal{L}_2(v)\} = 0, \\ v(T, S_1, S_2) = H_2(S_1, S_2)\,. \end{cases} \tag{58}$$

Similar to the previous problems, the loss function is built as the sum of the interior and boundary loss functions as follows:

$$\begin{aligned} L(v) =& \lambda L_I(v) + (1-\lambda) L_B(v) \\ =& \lambda \int_\Omega |\, \max(H_2(\mathbf{x}) - v, \mathcal{L}(v))\,|^p \, d\Omega + \\ & (1-\lambda) \int_{\partial\Omega} (|\, v(t,\mathbf{x}) - G(t,\mathbf{x})\,|^p + |\, v(t,\mathbf{x}) - H(\mathbf{x})\,|^p) d\gamma, \end{aligned} \tag{59}$$

where function G refers to the boundary conditions and function H denotes the final condition imposed for the problems. Note that the loss function is a generalization of the loss function introduced for the one asset problem and the integral terms are also approximated by Monte Carlo techniques.

3. ANN Option Pricing Results

After introducing the PDE model and the methodology, the European and American options values are computed with the ANNs based on the loss functions introduced. The unsupervised learning methodology from the previous section is adopted to compute the solutions and show some results. For the following tests, we have considered the parameter $p = 2$ in the loss functions.

3.1. European Options

First of all, a discussion is made about the European option single asset results obtained solving the PDE problem (12) by ANNs.

The results are presented with the loss function introduced in (19) and here the basic choice $\lambda = 0.5$ is used. Recall that optimal loss weight-based loss function or the variance normalization technique are especially useful in the case of nontrivial solutions.

The first example is a European put option, with the following parameters values: $\sigma = 0.25$, $r = 0.04$, $T = 1$, $K = 15$, $S_\infty = 4K$, $\delta = 0.0$. In Figure 1, the ANN-based, trained and the analytical solution are plotted for two time instances.

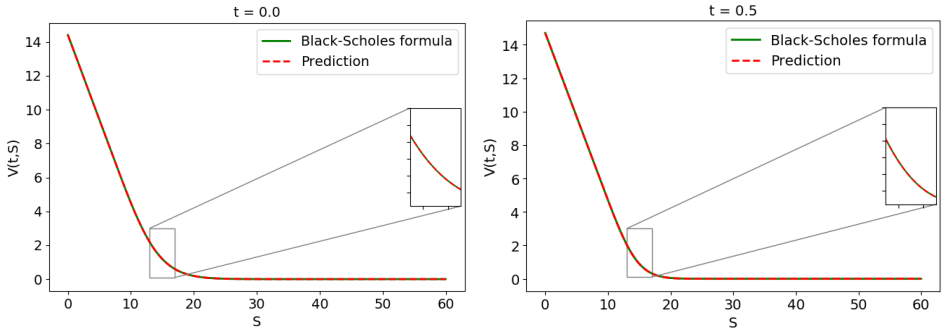

Figure 1. European put option for different time instances, $t = 0, t = 0.5$, with $\lambda = 0.5$.

The accuracy of the solution generated by the ANN is measured by comparing the relative error of the trained solution v_{ANN} with the analytic solution v_{BS}, as follows:

$$error = \frac{\|v_{BS} - v_{ANN}\|_{L^2}}{\|v_{BS}\|_{L^2}}. \tag{60}$$

In Figure 2 the error throughout the domain is plotted. Clearly, the biggest error in the ANN solution is found close to the strike price at maturity time $t = T$, where the payoff is non-smooth. The relative error according to (60) with $\lambda = 0.5$ is equal to 2.23×10^{-4}.

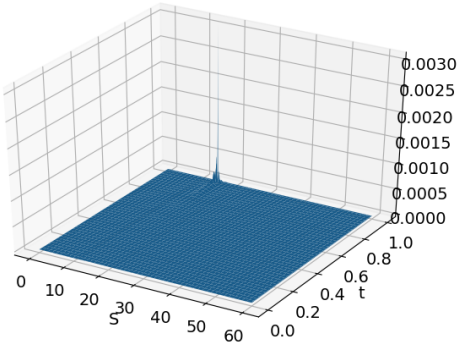

Figure 2. Error surface for the artificial neural network (ANN) solution.

Next, some results for a European option depending on two underlying assets are shown.

The corresponding loss function has been optimized by means of the L-BFGS algorithm and choosing the *tanh* as the activation function. In the last layer a linear activation function

is considered. In Figure 3 the ANN solution for the European exchange option has been plotted. The error, comparing the approximated ANN solution with the analytical solution given by (28) is also plotted. Note that the maximum error is obtained for the minimum value of both asset prices, which is related to S_1/S_2 in the expression of d_1 in (28).

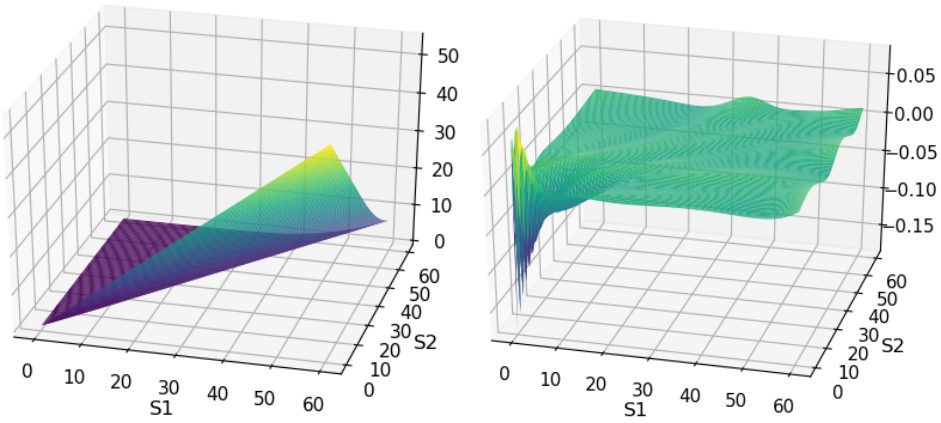

Figure 3. European exchange option (**left**), with parameters: $\sigma_1 = \sigma_2 = 0.25$, $\rho = 0.1$, $r = 0.05$, $\delta_i = 0.1$, $S_{1\infty} = S_{2\infty} = 60$ and loss weight $\lambda = 0.5$. The error surface, comparing with the analytical solution (**right**).

Due to the relatively big differences in the asset prices S_1, S_2 and the time t-values, we have scaled the inputs of the artificial neural network, i.e., the original domain $\widetilde{\Omega} = [0, S_{1\infty}] \times [0, S_{2\infty}]$, is scaled to a dimensionless computational domain, i.e., $\widetilde{\Omega}^* = [0,1] \times [0,1]$. Note that the size of the domain changes with the strike value, however the computational domain is always scaled, so that the solution is the option price in $(S_1, S_2) \in \widetilde{\Omega}$. By pricing the option with the parameters, $S_{1\infty} = S_{2\infty} = 4K$, $\sigma_1 = \sigma_2 = 0.25$, $\rho = 0.1$, $r_R = 0.04$, $r = 0.3$ and $T = 1$, for several values of K, modifying the original domain, it is found that scaling the input parameters is not sufficient to obtain highly accurate results for large domain sizes. In Table 1, the error for a European max-call option is presented, based on different unscaled domain sizes. It can be observed that as the domain increases the accuracy of the neural network solution decreases.

Table 1. Relative error for different domains.

$(S_{1,\infty}, S_{2,\infty})$	**Relative Error**
(10, 10)	2.58×10^{-4}
(60, 60)	3.17×10^{-4}
(120, 120)	8.08×10^{-4}
(180, 180)	1.71×10^{-2}
(240, 240)	2.75×10^{-1}
(300, 300)	3.96×10^{-1}
(360, 360)	4.30×10^{-1}

In order to understand the reasons for the degraded accuracy with an increasing domain size, the gradients of the interior and boundary loss functions have been computed. In Table 2, these values are presented for the European max-call. The gradient of the interior loss remains constant, note that the domain is always $[0,1] \times [0,1]$, however, the gradient of the boundary loss increases with the size domain. Clearly, the interior and boundary loss functions do not have the same dependency on the domain size.

Table 2. Gradient values for different domain sizes with standard weights.

$(S_{1,\infty}, S_{2,\infty})$	$\|\partial L_I/\partial\omega\|_{L^2}$	$\|\partial L_B/\partial\omega\|_{L^2}$	$\|\partial L_I/\partial\omega\|_{L^2}/\|\partial L_B/\partial\omega\|_{L^2}$
(10, 10)	0.4325	8.8515	0.04886
(60, 60)	0.4325	52.6274	0.00821
(120, 120)	0.4325	105.1598	0.00411
(180, 180)	0.4325	157.6923	0.00274
(240, 240)	0.4325	210.2249	0.00205
(300, 300)	0.4325	262.7574	0.00164
(360, 360)	0.4325	315.2899	0.00137

The goal is to compute accurate approximations of the solution independent the domain size, and therefore the ANN needs to be modified. The initialization of the weights is adapted by using a variation of the Xavier initialization. In particular, the initial values of the weight values in the last layer of the ANN will be scaled, by multiplying them by the maximum option value. As a result, a solution which is accurate independent of the size of the domain is obtained, see Table 3. This adaptation, i.e., the weights having similar magnitude as the expected largest option value in the output, forms a robust weight initialization. Moreover, such initialization helps for the interior and boundary loss functions to have similar sensitivity to the domain size. In Table 4, such behaviour can be observed, where the rate between both gradients remains close to $1/3$ when the size of the domain increases. Our results show that the BFGS optimization doesn't seem to pick up the gradient if the initial weights are not sufficiently large. Moreover, similar results can be observed when the inputs are not scaled.

Table 3. Relative error with scaled weights.

$(S_{1,\infty}, S_{2,\infty})$	**Relative Error**
(10, 10)	3.60×10^{-4}
(60, 60)	3.19×10^{-4}
(120, 120)	3.56×10^{-4}
(180, 180)	4.00×10^{-4}
(240, 240)	2.65×10^{-4}
(300, 300)	3.14×10^{-4}
(360, 360)	4.29×10^{-4}

Table 4. Gradient value for different domains with scaled weights.

$(S_{1,\infty}, S_{2,\infty})$	$\|\partial L_I/\partial\omega\|_{L^2}$	$\|\partial L_B/\partial\omega\|_{L^2}$	$\|\partial L_I/\partial\omega\|_{L^2}/\|\partial L_B/\partial\omega\|_{L^2}$
(10, 10)	26.372	74.790	0.35261
(60, 60)	949.424	2692.448	0.35262
(120, 120)	3797.696	10,769.792	0.35262
(180, 180)	8544.816	24,232.031	0.35262
(240, 240)	15,190.74	43,079.168	0.35262
(300, 300)	23,735.602	67,311.2	0.35262
(360, 360)	34,179.266	96,928.125	0.35262

Based on the adapted weights initialization, in Table 5, the results for the European max-call option are presented, and the solution computed by the ANN is compared with the analytical solution given by (33) for some specific asset prices and different strike values, based on the corresponding loss function and $\lambda = 0.5$. The parameters considered are, $\sigma_1 = \sigma_2 = 0.2$, $\rho = 0.1$, $\delta_i = 0.1$, $r = 0.05$ and $T = 1$. Moreover, the maximum value of the asset prices is $S_{1\infty} = S_{2\infty} = 4K$. Note that the accuracy of the trained solution is not affected by the size of the domain, in addition, similar to the one-dimensional case, the maximum error is obtained when the underlying value is close to the strike price.

Table 5. European max-call option value.

Strike	(S_1, S_2)	ANN	Analytical
15	$(15, 15)$	-3.16×10^{-2}	8.90×10^{-5}
	$(10, 20)$	4.58×10^{-2}	1.16×10^{-2}
	$(25, 5)$	2.02×10^{-1}	2.11×10^{-1}
30	$(30, 30)$	-4.23×10^{-2}	1.78×10^{-4}
	$(20, 40)$	4.75×10^{-2}	2.32×10^{-2}
	$(50, 10)$	4.10×10^{-1}	4.21×10^{-1}
60	$(60, 60)$	1.035×10^{-1}	3.56×10^{-4}
	$(40, 80)$	1.79×10^{-1}	4.63×10^{-2}
	$(100, 20)$	7.79×10^{-1}	8.42×10^{-1}

In Table 6, the error for the two-asset European options is presented. The values are computed based on the expression in (60) and very fine accuracy for the problems is observed.

Table 6. Error according to the loss weight values.

Option	λ	Error
Asset exchange	0.5	4.16×10^{-4}
Max-call $K = 15$	0.5	4.55×10^{-4}
Max-call $K = 30$	0.5	3.51×10^{-4}
Max-call $K = 60$	0.5	3.83×10^{-4}

3.2. American Options

The goal of this section is to address the American option problem by using unsupervised learning with the ANN. As for the European option, the values for one and two underlying assets are computed. However, whereas for the European option an analytical solution is known, for the American case, the reference values will be the option values computed by finite elements (FEM) using the numerical methods in [30,31] to solve the linear complementarity problem for the American options.

Similar to the European options problem, the loss function has been optimized using the L-BFGS algorithm, moreover, the ANN is based on the activation function *tanh(x)*. With the aim of comparing both methodologies, an American option is priced with the same parameter data set as in the previous example, considering now, for one underlying asset the optimal loss weight, which equals $\lambda \approx 0.90$. First of all, American options are determined with the following parameter data: $\sigma = 0.25$, $\delta = 0.26$, $r = 0.3$, $T = 1$, $K = 15$, $S_\infty = 4K$. Figure 4 shows the trained solution and the error related to a reference FEM solution, for all time points. As for the European options, the maximum error is reached when the asset price is equal to the strike price, close to the maturity time, where the payoff function is not smooth. In Figure 5, a comparison of the American option value computed

by ANNs or FEM is presented for different time points. Moreover, the payoff is plotted to demonstrate that the obstacle condition is satisfied.

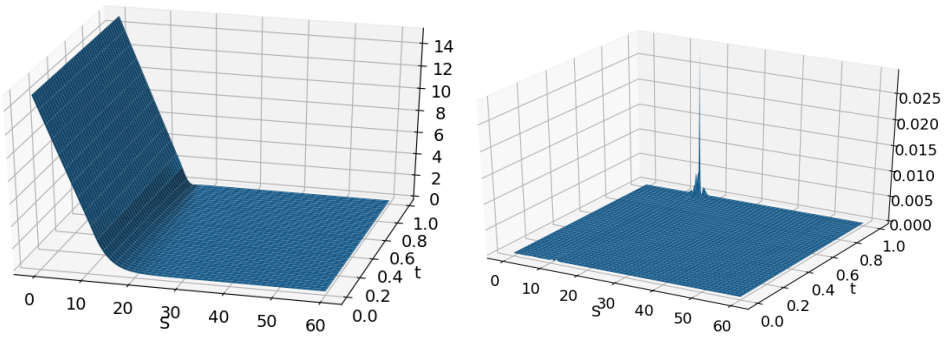

Figure 4. American option price with dividends (**top**). Error surface comparing with the solution obtained by finite element method (**bottom**). Solution obtained by variance normalization method.

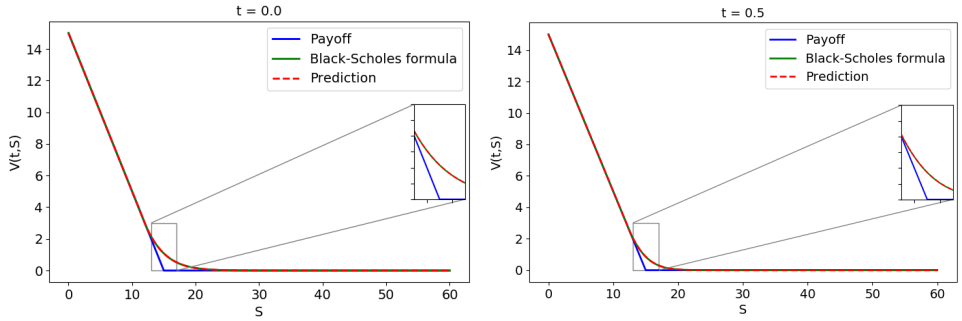

Figure 5. American price and the payoff function. Finite element method (green line) and Neural networks (red line) for different time points. Solution obtained by variance normalization method.

The accuracy of two loss functions for the LCP, one based on optimal loss weight and another based on variance normalization, is compared by means of the relative error of the solution, computed in terms of the L^2-norm, similar to (60), i.e.,

$$error = \frac{\|v_{FEM} - v_{ANN}\|_{L^2}}{\|v_{FEM}\|_{L^2}}. \tag{61}$$

Very similar accuracy is obtained with both loss functions, 5.38×10^{-4} (for optimal loss weight) versus 5.82×10^{-4} (variance normalization). However, comparing the convergence of both methodologies, which is presented in Figure 6, it can be clearly observed that by defining the loss function with a variance normalization (top) the neural network converges faster than using the optimal loss weight (bottom). In this figure, the relative error is plotted for different numbers of iterations of the L-BFGS algorithm.

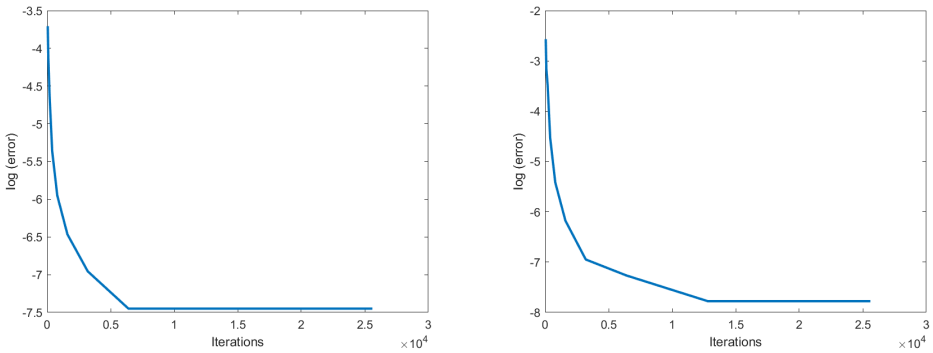

Figure 6. Error value (represented in log scale) obtained for different iterations with the variance normalization method (**top**) and using the optimal loss weight (**bottom**). The reference solution has been obtained solving the PDE by the finite element method.

Next, the American options depending on two underlying assets are valued. Optimizing the loss function with the L-BFGS algorithm, with the *tanh(x)* as the activation function, and equal weighting of boundary and interior losses, with $\lambda = 0.5$, the following results have been obtained for the three types of options.

In Table 7, a comparison between the ANN and FEM solutions is shown. An American max-call option has been studied with several strike values and the following parameter data, $\rho = 0.1$, $\sigma_1 = \sigma_2 = 0.25$, $r = 0.04$, $\delta = 0.01$ and $T = 0.5$. Moreover, for the FEM, 75 time steps have been considered for the time discretization and the spatial discretization is based on a 101×101 mesh.

Table 7. Comparison of American max-call option values.

Strike	(S_1, S_2)	ANN	FEM
15	(15, 15)	2.021	2.066
	(10, 20)	5.703	5.643
	(25, 5)	10.996	10.969
30	(30, 30)	4.102	4.133
	(20, 40)	11.405	11.29
	(50, 10)	21.998	21.938
60	(60, 60)	7.916	8.266
	(40, 80)	22.753	22.573
	(100, 20)	43.994	43.877

Figure 7 shows the trained solution for the spread American option with the following parameter values, $K = 15$, $S_{1\infty} = S_{2\infty} = 4K$, $\sigma_1 = \sigma_2 = 0.25$, $\rho = 0.0$, $r_R = 0.04$, $r = 0.3$ and $T = 1$, and the error surface using the FEM solution following [31] as a reference. In Figure 8, the option value and the difference with the payoff function are shown.

Finally, in order to show the accuracy of the method applied to train the ANN to price American options depending on two asset prices, the relative error is presented in Table 8.

Note that the accuracy of the neural network for the American options depending on two stochastic factors is lower than for the European options. However, this may be because here a numerical solution is our reference and not a closed-form expression.

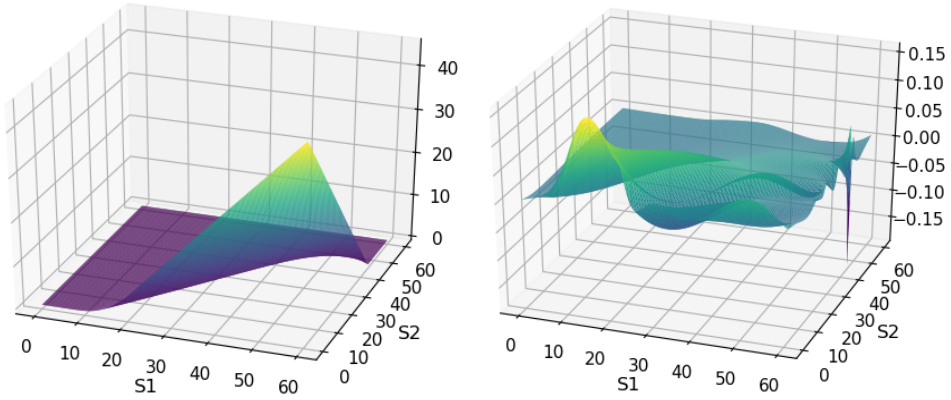

Figure 7. Two-asset spread American option value in the whole domain (**left**). Error surface between the FEM and the ANN solution (**right**).

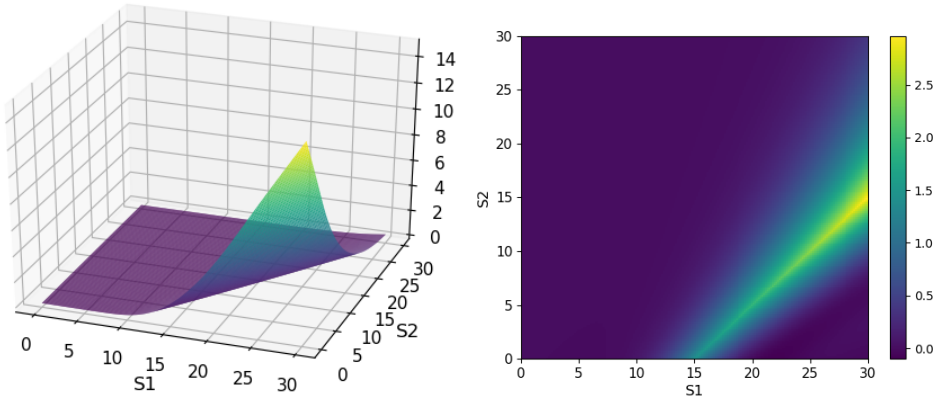

Figure 8. Two-Asset spread American option value in a reduced domain (**left**). Difference between the ANN solution and the payoff function (**right**).

Table 8. Error for different multi-asset American options.

	λ	**Error**
Max-call	0.5	1.73×10^{-3}
Spread	0.5	2.45×10^{-3}
Arithmetic average put	0.5	6.42×10^{-3}

4. Conclusions

In this work, classical problems in financial option pricing have been addressed with artificial neural networks. In particular, following the classical Black–Scholes model, European and American options depending on one and two underlying assets have been valued. A new unsupervised learning methodology is introduced to solve the option value problems based on the PDE formulation. With this aim, we proposed appropriate loss functions. The classical Black–Scholes American option pricing problem has been formulated as a linear complementarity problem.

For the European option problem, the accuracy of the methods was compared to the analytical solution, whereas, for American options, solutions computed by the finite element method were used as reference values. For all problems considered, the final error in the ANN solution was highly satisfactory. So, in this work a new, different methodology to price financial options has been proposed, based on unsupervised learning with ANNs. As a result, highly satisfactory, accurate PDE solutions are obtained. Needless to mention that ANNs can be easily extended to solving higher-dimensional problems, as they are not drastically affected by the curse of dimensionality. When solving PDEs by unsupervised

learning, boundary conditions need to be incorporated and weighted by means of a suitable loss function.

For future work, different asset price models can be chosen, like the Heston stochastic volatility model instead of the Black–Scholes model. Moreover, this methodology can be applied to price other financial derivatives, such as barrier or Asian options. In addition, the PDE problem formulation can be easily generalized by introducing counterparty risk which gives rise to nonlinear option valuation PDEs.

Author Contributions: Investigation and writing—original draft preparation: B.S., Supervision: C.W.O., validation: R.v.d.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: B. Salvador was funded by European Research Consortium for Informatics and Mathematics (ERCIM) fellowship.

Conflicts of Interest: Declare conflicts of interest or state "The authors declare no conflict of interest". "The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results".

References

1. Zurada, J.M. *Introduction to Artificial Neural Systems*; West Publishing Company: Egan, MN, USA, 1992.
2. Yadav, N.; Yadav, A.; Kumar, M. *Neural Network Methods for Solving Differential Equations*; Springer: Berlin, Germany, 2015.
3. Kryzanowski, L.; Galler, M.; Wright, D.W. Pricing of high-dimensional American options by neural networks. *Financ. Anal. J.* **1993**, *49*, 21–27. [CrossRef]
4. Refenes, A.P.; Zapranis, A.; Francis, G. Modelling stock returns in the framework of APT. *Neural Netw. Cap. Mark.* **1995**, *7*, 101–125.
5. Dutta, S.; Shekar, S. Bond rating: A non-conservative application of neural networks. *Proc. IEEE Int. Conf. Neural Netw.* **1988**, *2*, 443–450.
6. Moody, J.; Utans, J. Architecture selection strategies for neural networks. In *Neural Networks in the Capital Markets*; John Wiley & Sons: New York, NY, USA, 1994; pp. 141–183.
7. Singleton, J.C.; Surkan, A.J. Bond rating with neural networks. In *Neural Networks in the Capital Markets*; John Wiley & Sons: New York, NY, USA, 1995; pp. 301–307.
8. Becker, S.; Cheridito, P.; Jentzen, A. Pricing and hedging American-style options with deep learning. *arXiv* **2019**, arXiv:1912.11060.
9. Becker, S.; Cheridito, P.; Jentzen, A.; Welti, T. Solving high-dimensional optimal stopping problems using deep learning. *arXiv* **2019**, arXiv:1908.01602.
10. Amilon, H. A Neural Network Versus Black–Scholes:A Comparison of Pricing and Hedging Performances. *J. Forecast.* **2003**, *22*, 317–335. [CrossRef]
11. Grohs, P.; Hornung, F.; Jentzen, A.; Wurstemberger, P.V. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. Memoirs of the American Mathematical Society. *arXiv* **2019**, arXiv:1809.02362v1. To appear.
12. Han, J.; Jentzenb, A.; E, W. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. USA* **2018**, *34*, 8505–8510. [CrossRef]
13. Raissi, M.; Karniadakis, G.E.; Perdikaris, P. Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations. *arXiv* **2017**, arXiv:1711.10561v1.
14. van der Meer, R.; Oosterlee, C.; Borovykh, A. Optimally weighted loss functions for solving PDEs with Neural Netwoks. *arXiv* **2020**, arXiv:2002.06269.
15. Mohammadi, R. Quintic B-spline collocation approach for solving generalized Black–Scholes equation governing option pricing. *Comput. Math. Appl.* **2015**, *69*, 777–797. [CrossRef]
16. Longstaff, F.A.; Schwartz, E.S. Valuing American options by simulation: A simple least-squares approach. *Rev. Financ. Stud.* **2001**, *14*, 113–147. [CrossRef]
17. Bhatia, G.S.; Arora, G. Radial Basis Function Methods for Solving Partial Differential Equations-A Review. *Indian J. Sci. Technol.* **2016**, *9*, 1–18.
18. Hon, Y.; Mao, X.Z. A Radial Basis Function Method For Solving Options Pricing Model. *Financ. Eng.* **1998**, *8*, 31–50.
19. Liu, S.; Oosterlee, C.W.; Bothe, S.M. Pricing options and computing implied volatilities using neural networks. *Risks* **2019**, *7*, 16. [CrossRef]
20. Kohler, M.; Krzyzak, A.; Todorovic, N. Pricing of highdimensional American options by neural networks. *Math. Financ.* **2010**, *20*, 383–410. [CrossRef]

21. Sirignano, J.; Spiliopoulos, K. DGM:*A Deep Learning Algorithm for Solving Partial Differential Equations*; Journal of Computational Physics: Amsterdam, The Netherlands, 2018.
22. Liu, S.; Leitao, A.; Borovykh, A.; Oosterlee, C. On calibration neural networks for extracting implied information from American options. *arXiv* **2020**, arXiv:2001.11786v1.
23. Margrabe, W. The value of an option to exchange one asset for another. *J. Financ.* **1978**, *33*, 177–186. [CrossRef]
24. Johnson, H. Options on the maximum or the minimum of several assets. *J. Financ. Quant. Anal.* **1987**, *22*, 277–283. [CrossRef]
25. Stulz, R. Options on the minimum or the maximum of two risky assets: Analysis and applications. *J. Financ. Econ.* **1982**, *10*, 161–185. [CrossRef]
26. Wilmott, P.; Howison, S.; Dewynne, J. *The Mathematics of Financial Derivatives: A Students Introduction*; Cambridge University Press: Cambridge, UK, 1996.
27. Ikonen, S.; Toivanen, J. Operator Spliting Methods for American option pricing. *Appl. Mat. Lett.* **2004**, *17*, 809–814. [CrossRef]
28. Oleinik, O.; Radkevic, E. *Second Order Equations with Nonnegative Characteristics Form*; AMS, Plenum Press: New York, NY, USA, 1973.
29. Fichera, G. On a unified theory of boundary value problems for elliptic–parabolic equations of second order. *Mathematika* **1963**, *7*, 99–122.
30. Arregui, I.; Salvador, B.; Vázquez, C. PDE models and numerical methods for total value adjustment in European and American options with counterparty risk. *Appl. Math. Comput.* **2017**, *308*, 31–53. [CrossRef]
31. Arregui, I.; Salvador, B.; Ševčovič, D.; Vázquez, C. PDE models for American options with counterparty risk and two stochastic factors: Mathematical analysis and numerical solution. *Comput. Math. Appl.* **2020**. [CrossRef]