Journal of
Risk and Financial Management

MDPI

Article

Pricing Multi-Asset Bermudan Commodity Options with Stochastic Volatility Using Neural Networks

Kentaro Hoshisashi [1,2,3] and Yuji Yamada [4,*]

[1] Department of Computer Science, University College London, London WC1E 6BT, UK; kentaro.hoshisashi.22@ucl.ac.uk
[2] Graduate School of Business Sciences, University of Tsukuba, Tokyo 112-0012, Japan
[3] Sumitomo Mitsui Banking Corporation, Tokyo 100-0005, Japan
[4] Faculty of Business Sciences, University of Tsukuba, Tokyo 112-0012, Japan
[*] Correspondence: yuji@gssm.otsuka.tsukuba.ac.jp

Abstract: It has been recognized that volatility in commodity markets fluctuates significantly depending on the demand–supply relationship and geopolitical risk, and that risk and financial management using multivariate derivatives are becoming more important. This study illustrates an application of multi-layered neural networks for multi-dimensional Bermudan option pricing problems assuming a multi-asset stochastic volatility model in commodity markets. In addition, we aim to identify continuation value functions for these option pricing problems by implementing smooth activation functions in the neural networks and evaluating their accuracy compared with other activation functions or regression techniques. First, we express the underlying asset dynamics using the multi-asset stochastic volatility model with mean reversion properties in the commodity market and formulate the multivariate Bermudan commodity option pricing problem. Subsequently, we apply multi-layer perceptrons in the neural network to represent the continuation value functions of Bermudan commodity options, wherein the entire neural network is trained using the least-squares Monte Carlo simulation method. Finally, we perform numerical experiments and demonstrate that applications of neural networks for Bermudan options in a multi-dimensional commodity market achieve sufficient accuracy with regard to various aspects, including changing the exercise dates, the number of layers/neurons, and the dimension of the problem.

Keywords: Bermudan commodity options; multi-layer perceptron; multi-asset stochastic volatility model

check for updates

Citation: Hoshisashi, Kentaro, and Yuji Yamada. 2023. Pricing Multi-Asset Bermudan Commodity Options with Stochastic Volatility Using Neural Networks. *Journal of Risk and Financial Management* 16: 192. https://doi.org/10.3390/jrfm16030192

Academic Editors: Kentaro Iwatsubo and Thanasis Stengos

Received: 28 December 2022
Revised: 5 March 2023
Accepted: 10 March 2023
Published: 12 March 2023

1. Introduction

In commodity markets, typical products include directional trades such as futures and forwards, which establish an obligation to purchase or sell an underlying commodity in the future (Clark 2014). As essential tools for managing risks from these contracts, which may consist of multiple underlying assets, there are various options contracts that provide a right to trade the underlying commodity under a specified condition. In this study, we focus on early-exercisable options on multiple underlying assets in commodity markets, i.e., multivariate Bermudan commodity options.

Solving Bermudan commodity option pricing problems with multiple underlying assets and factors is challenging because computational efforts grow exponentially in tandem with the problem dimension in general, which is determined by the number of assets and factors. However, the improvement of algorithms and the rapid growth of computational power have led to a remarkable surge of interest in computational science in recent years. Currently, a wide variety of machine learning algorithms, such as deep learning and neural networks, are successfully employed for classification, regression, clustering, or dimensionality reduction tasks and are applied for large-sized and high-dimensional data in various areas. In this study, we develop a new Bermudan commodity option algorithm via multi-layered neural networks and show its efficiency and effectiveness based on the multi-asset commodity market model with stochastic volatility, wherein significant changes in volatility may be observed according to the demand–supply relationship and geopolitical conditions in commodity markets.

An important feature of Bermudan options is that they can be exercised early, with their value being determined by whether or not they are exercised before maturity. In other words, the option holder must decide whether to continue holding the option or immediately exercise it at a prespecified period. In this situation, it is crucial to determine the continuation value, i.e., the value of holding the option until the next exercise window. Such a continuation value may be given as the discounted conditional expectation of the remaining option value on one step ahead under a risk-neutral probability measure, which generally has no explicit boundary conditions. Moreover, the conditional expectation providing the continuation value is an unknown (possibly nonlinear and complex) multivariate function whose dimensions depend on the number of underlying factors; hence, an exact (yet still approximate) computation involves high-dimensional discrete grids concerning state variables and is quite difficult to solve.

To price early-exercisable options with estimations of continuation value functions, Longstaff and Schwartz (2001) proposed a simple yet powerful numerical method involving a regression-based functional estimation using simulated sample paths known as the least-squares Monte Carlo (LSMC) method. Since then, several studies have examined the application of neural networks (or machine learning methods) for estimating continuation value functions in option pricing based on the LSMC method. For American options' pricing, Haugh and Kogan (2004) applied a neural network with one hidden layer for valuation, whereas Kohler et al. (2010) proved price consistency and convergence with multiple payoff types. Lapeyre and Lelong (2021) gave several numerical examples of Bermudan options and proved convergence. There are additional examples, e.g., the 5000 assets rainbow option (Becker et al. 2021) and expected exposures (Andersson and Oosterlee 2021). Furthermore, other machine learning algorithms have been used for early-exercisable options, e.g., radial basis functions (Ballestra and Pacelli 2013), nearest-neighbor (Feng et al. 2013), deep learning (Becker et al. 2020; Liang et al. 2021), unsupervised learning (Salvador et al. 2020), and reinforcement learning (Li 2022), as well as the support vector machine (Lin and Almeida 2021). Moreover, numerical approaches have also been used, including stochastic kriging metamodels (Ludkovski 2018), high-dimensional partial differential equations (Sirignano and Spiliopoulos 2018), and backward stochastic differential equations (Chen and Wan 2021). Furthermore, there are other applications of neural networks in the finance field, e.g., extending the feature set (Montesdeoca and Niranjan 2016), the calculation of implied volatilities (Liu et al. 2019), and decision-making (Puka et al. 2021). A comprehensive review of these methods was conducted by Ruf and Wang (2020).

Although this study shares the same ideas as the aforementioned studies—in particular, as in Lapeyre and Lelong (2021), given that a multi-layer perceptron (MLP) is applied in the neural network—it is worthwhile to mention that our study may be considered novel in several aspects: We illustrate an algorithm for estimating the continuation values of multi-asset Bermudan commodity options with stochastic volatility features, whereby a smooth activation function, such as the sigmoid function, is applied in the MLP to reflect the smoothness of conditional expectations regarding state variables. The smoothness of functions to represent conditional expectations is key in this study. In the Markovian setting, the conditional probability density functions are usually smooth given state variables; thus, conditional expectations are smooth functions. Therefore, the target continuation value function is smooth, and we can expect a better fit to the target function by using a smooth activation function in the MLP. This is in contrast with more commonly used piecewise linear functions such as the leaky ReLU function applied in the numerical experiments by Lapeyre and Lelong (2021), wherein the fitting function may not be smooth but only piecewise smooth. (Note that, in a similar context with the smoothness of estimated functions, Yamada (2012, 2017) applied the generalized additive model to calculate smooth functions for conditional expectations in multivariate hedging problems with European and Bermudan options.) Additionally, we applied more sophisticated techniques such as the resampling procedure and early stopping to improve the computational efficiency and avoid possible biases in pricing or overfitting for optimal learning (see Section 3.2 and the numerical experiments).

While several neural network/machine learning models for option pricing exist, we believe that, for this type of research, the current methodologies, along with developed computational algorithms, need to be combined with existing techniques using the currently available computational environment. In this context, the choice of problem and methodology choice is important, as is the approach to the problem and how to perform the numerical experiments. The combination of multi-asset commodity options with stochastic volatility and recently developed neural network techniques (including the computational environment and software) is meaningful since commodity markets are largely volatile, and this volatility may change over time. Moreover, the multi-asset Bermudan commodity options with stochastic volatility, to the best of our knowledge, have not been previously considered despite the problem's importance. It should become more challenging in numerical calculations to configure multiple underlying assets that recognize mean-reverting dynamics and solve boundary conditions with stochastic factors (see, e.g., Hahn and Dyer 2008 and Ball and Roma 1994).

The present study implements a multi-layered neural network and examines its efficiency and effectiveness for multi-asset Bermudan commodity option pricing problems with stochastic volatility. First, we formulate the multi-asset commodity market with stochastic volatility, wherein individual asset price dynamics are expressed as a two-factor model by combining a well-known commodity model by Schwartz (1997) with Heston's stochastic volatility model (Heston 1993). Next, we apply MLPs in the neural network to represent the continuation value functions in Bermudan option pricing, whereby the entire neural network is trained using LSMC simulations. We perform numerical experiments to compare the continuation value function accuracy in response to changing the exercise dates, the number of layers/neurons, and the dimension of the problem. We also compare the relationship between the continuation values and network configurations.

The outline of this article is as follows. Section 2 gives an introduction to the commodity option structure adopted in this study and the formulation of the multi-dimensional Bermudan option problem. Section 3 describes the configuration of neural networks, a multi-dimensional asset model with stochastic volatility, and a Bermudan options pricing procedure for learning and valuation via Monte Carlo sample paths. Section 4 presents the numerical results of the Bermudan option prices and compares the accuracy of the continuation value surfaces. Section 5 summarizes the analysis results and discussions. Lastly, Section 6 concludes this study.

2. Pricing Multi-Asset Bermudan Commodity Options with Stochastic Volatility

In this section, we introduce early-exercisable commodity options and formulate the problem of pricing multi-asset Bermudan commodity options with stochastic volatility.

2.1. Early-Exercisable Commodity Options

As stated earlier, in commodity markets, typical products include plain directional trades such as future and forward contracts, which establish an obligation to buy or sell a particular commodity asset at a specified price in the future (Clark 2014). Depending on the terminal values of commodity assets, holding these contracts may lead to a loss or profit for the contractor, while a large loss is particularly undesirable for the holder; furthermore, the possibility of a large profit may be pursued. Such opportunities are realized using options contracts, giving a right to purchase or sell an underlying commodity asset with a prespecified strike price in the future.

Among the many types of options used as hedging tools in commodity markets, early-exercisable options provide additional flexibility regarding exercise timing and are considered useful for hedgers, in practice. Traditionally, such options are characterized as American options; however, in the context of exotic options, Bermudan options have similar flexibility. These options allow holders to exercise them early, although only on specific dates before maturity; thus, the option holder must decide whether to continue holding the option or immediately exercise it during the exercisable period. However, such continuation value is usually unknown because it depends on future option values on specific exercisable dates. Therefore, it is paramount to determine the continuation values for Bermudan options. The objective of this study is to evaluate computational performance (including the accuracy of continuation value estimation) for pricing multi-asset Bermudan commodity options via multi-layered neural networks.

2.2. Multi-Dimensional Bermudan Option Pricing Problem

In this subsection, we describe the multi-dimensional Bermudan option pricing problem, following Lapeyre and Lelong (2021). Given a complete filtered probability space (Ω, $\mathcal{F}$, $(\mathcal{F}_t)_{0\leq t\leq T}$, $\mathbb{P}$) with a finite time horizon $T > 0$, we assume that a set of underlying assets is modeled via a multifactored process $(X_t)_{0\leq t\leq T}$ adapted to the filtration, $(\mathcal{F}_t)_{0\leq t\leq T}$, and that $\mathbb{P}$ is an associated risk-neutral measure. We consider a Bermudan option with exercise dates $0 = T_0 \leq T_1 < T_2 < \ldots < T_N = T$ and a discrete-time payoff process P_{T_n} if exercised at times $(T_n)_{0\leq n\leq N}$, where P_{T_n} is specified as a function of X_{T_n}. Then, Bermudan option prices Z_{T_n} are computed using the following recursive equation:

$$\begin{cases} Z_{T_N} = P_{T_N} \\ Z_{T_n} = max\left(P_{T_n}, e^{-r\delta_{T_n}}\mathbb{E}\left[Z_{T_{n+1}}|\mathcal{F}_{T_n}\right]\right),\ 0 \leq n \leq N-1 \end{cases} \tag{1}$$

where $\mathbb{E}$ denotes the expectation under the risk-neutral probability measure $\mathbb{P}$ with the risk-free interest rate r and the interval between T_{n-1} and T_n as δ_{T_n}. Furthermore, assuming that $(X_t)_{0\leq t\leq T}$ is a multi-dimensional Markov process, there exists a measurable function $\Phi_n : \mathbb{R}^{d_x} \to \mathbb{R}$, such that:

$$e^{-r\delta_{T_n}}\mathbb{E}\left[Z_{T_{n+1}}|\mathcal{F}_{T_n}\right] = e^{-r\delta_{T_n}}\mathbb{E}\left[Z_{T_{n+1}}|X_{T_n}\right] = \Phi_n(X_{T_n}),\ \ 0 \leq n \leq N-1. \tag{2}$$

Herein, we refer to Φ_n as a continuation value function in this paper.

Note that finding the exact Φ_n is difficult; alternatively, one may identify a function f_n to minimize the following quantity,

$$\mathbb{E}\left[\left|e^{-r\delta_{T_n}}Z_{T_{n+1}} - f_n(X_{T_n})\right|^2\right], \tag{3}$$

over a parametrized set of functions $\mathfrak{B}$. If all (real-valued) square-integrable measurable functions are searched to minimize Equation (3), it turns out that the function Φ_n providing the conditional expectation in Equation (2) is achieved via an optimizer. However, there is a trade-off between the generality of a set of functions and the efficiency of computation. Additionally, computational tractability depends on the methodology to solve the optimization problem.

2.3. Multi-Asset Commodity Market Model with Stochastic Volatility

This study employs a multivariate commodity market model consisting of multiple underlying assets with stochastic volatility for the Bermudan option problem. To this end, we adopt a stochastic volatility model for the mean-reverting commodity dynamics (Schwartz 1997) and expand it to the multi-asset case.

Consider the Bermudan option problem with n underlying assets at time t, $S_{i,t}$, $i = 1, \ldots, l$, the i-th price dynamics of which are governed by the following two-dimensional stochastic differential equations (SDEs):

$$\begin{aligned} dS_{i,t} &= \kappa_{S_i}(\mu_i - \ln S_{i,t})S_{i,t}dt + \sqrt{v_{i,t}}S_{i,t}\,dW_{S_i,t}, \\ dv_{i,t} &= \kappa_{v_i}(\theta_i - v_{i,t})dt + \xi_i\sqrt{v_{i,t}}dW_{v_i,t}. \end{aligned} \tag{4}$$

Herein, $W_{S_i,t}$ and $W_{v_i,t}$ are correlated Brownian motions with appropriate correlation parameters, while the magnitude of the speed coefficient κ_{S_i} measures the degree of mean reversion to the long-run mean μ_i, including the market price of risk in the underlying asset price processes. The second term characterizes the i-th volatility process, $\sigma_{i,t} \equiv \sqrt{v_{i,t}}$, where κ_{v_i} indicates a degree of mean reversion toward long-term volatility θ_i, and ξ_i is the volatility of volatility.

Since each of the underlying asset price dynamics in (4) follows a two-dimensional Markov process, the state variables at time t, denoted by X_t, corresponding to the input features X of the MLP in the previous subsection, may be described as

$$X_t := [S_{1,t}, \sigma_{1,t},\ S_{2,t}, \sigma_{2,t},\ \ldots, S_{l,t}, \sigma_{l,t}]^\top \in \mathbb{R}^{d_l}. \tag{5}$$

The dimension d_l in (5) depends on the number of state variables and is given by $d_l := 2l$. Note that the SDEs of the underlying assets are used to generate sample paths of the LSMC method in the Bermudan option pricing problem.

3. Application of Neural Networks with MLP

When pricing Bermudan commodity options using a model with multi-dimensional factors—including the multi-asset stochastic volatility model introduced in the previous section—it is crucial to determine continuation values at each exercisable date. In order to identify a continuation value function in the multi-asset Bermudan option pricing problem with stochastic volatility, this study takes a neural network approach with MLP, similar to Lapeyre and Lelong (2021). First, we introduce the neural network architecture considered in this study, which generates a continuation value in Bermudan commodity options pricing. Second, we present the underlying assets model with multi-dimensional factors, which has multi-asset and stochastic volatility. Finally, we provide algorithms for learning the entire network and option pricing procedure.

3.1. Continuation Value Functions via MLPs

First, we explain the configuration of an MLP to express a general multi-dimensional function and to approximate the continuation value function in the multi-dimensional Bermudan option problem.

The basic configuration of the MLP is shown in Figure 1, where $X \in \mathbb{R}^d$ is an input vector and $Z \in \mathbb{R}$ is an output of the entire neural network. Each neuron is called a "perceptron" that defines a mapping of input/output signals with appropriate dimensions, being dependent on the number of neurons at input/output layers. For example, if $\boldsymbol{x} \in \mathbb{R}^{d_x}$ denotes an input signal of a perceptron with a weight matrix $\boldsymbol{W} \in \mathbb{R}^{d_x \times d_y}$ and a bias vector $\boldsymbol{b} \in \mathbb{R}^{d_y}$, then, an output signal $\boldsymbol{y} \in \mathbb{R}^{d_y}$ from the perceptron is given by

$$\boldsymbol{y} = h\left(\boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b}\right), \tag{6}$$

where $h : \mathbb{R}^{d_y} \to \mathbb{R}^{d_y}$ is a component-wise activation function. Typical choices of activation functions in neurons are as follows:

$$\begin{array}{c} \text{Sigmoid}: \ x \mapsto \frac{1}{1+e^{-x}} \\ \text{ReLU}: x \mapsto \max(x,\ 0) \end{array}. \tag{7}$$

In the case where the MLP is applied for a regression, all the weight matrices and bias vectors in the MLP are computed to minimize the sum of squared errors between the actual dependent variable, denoted by $Z \in \mathbb{R}$, and the predicted dependent variable $\hat{Z} \in \mathbb{R}$ given the training datasets of $X \in \mathbb{R}^{d_x}$ and $Z \in \mathbb{R}$ (expressed using the MLP in Figure 1).

Note that the MLP can express a continuous and complex nonlinear surface in entire networks by sequentially performing a linear and nonlinear transformation on inputs $X \in \mathbb{R}^{d_x}$ to the compiled layer output $\hat{Z} \in \mathbb{R}$. The properties of the MLP function derive from the universal approximation theorem proposed by Cybenko (1989) and the Kolmogorov–

Arnold representation theorem put forward by Kolmogorov (1957) and Arnold (2009), in which any function can be approximated if the input size and network are infinite. In this sense, functions expressed by the MLP are generally considered suitable for a problem with complicated interactions because of the adjustable basis functions (see Choon et al. 2008).

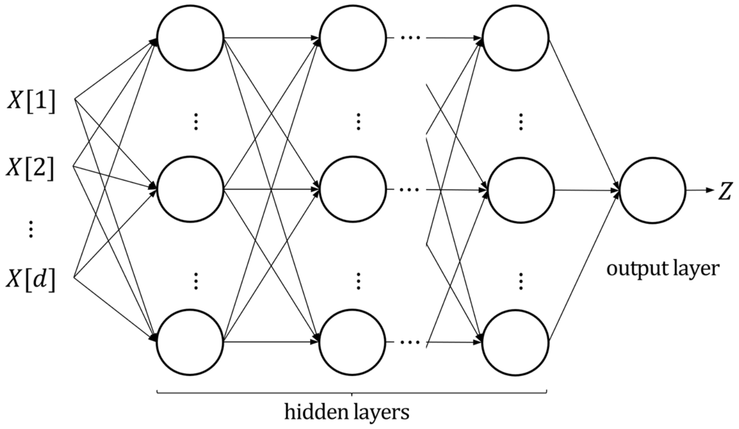

Figure 1. Multi-layered structure with perceptrons (circles) in a multi-layer perceptron.

For the randomly generated sample paths of $(X_t)_{0\leq t\leq T}$, we can apply the LSMC method (see Appendices A and B) combined with the MLP, whereby the continuation value function is modeled at each step using a function given by the MLP. Let $X_t^{(1)}, X_t^{(2)}, \dots, X_t^{(M)}$ and $0 \leq t \leq T$ be the simulated M sample paths of $(X_t)_{0\leq t\leq T}$. Since the discrete-time payoff process P_{T_n} (if exercised at times $(T_n)_{0\leq n\leq N}$) is specified as a function of X_{T_n} for a Bermudan option and $Z_{T_N} = P_{T_N}$, the training data of the output variable, $Z \equiv Z_{T_N} \in \mathbb{R}$, in the first step of the LSMC method, are computed as $P_{T_N}^{(1)}, P_{T_N}^{(2)}, \dots, P_{T_N}^{(M)}$, corresponding to the payoffs of the Bermudan option at maturity along the sample path of X_{T_N}. The MLP in the first step is constructed for the training data of $X \equiv X_{T_{N-1}} \in \mathbb{R}^{d_x}$, given as $X_{T_{N-1}}^{(1)}, X_{T_{N-1}}^{(2)}, \dots, X_{T_{N-1}}^{(M)}$, together with those of $Z \equiv Z_{T_N} \in \mathbb{R}$. Then, we obtain an approximation of the continuation value function, denoted by $\hat{\Phi}_{N-1}$, and the continuation values along the sample path, $\hat{\Phi}_{N-1}\left(X_{T_{N-1}}^{(m)}\right)$, $m = 1, \dots, M$.

In the second step, the training data of the output variable, $Z \equiv Z_{T_{N-1}} \in \mathbb{R}$, are computed using (1), as

$$Z_{T_{N-1}}^{(m)} = max\left(P_{T_{N-1}}^{(m)},\ \hat{\Phi}_{N-1}\left(X_{T_{N-1}}^{(m)}\right)\right),\ m = 1, \dots,\ M, \tag{8}$$

as well as the training data of $X \equiv X_{T_{N-2}} \in \mathbb{R}^{d_x}$, given as $X_{T_{N-2}}^{(1)}, X_{T_{N-2}}^{(2)}, \dots, X_{T_{N-2}}^{(M)}$. Using these training datasets, the MLP is constructed to find an approximation of the continuation value function, denoted by $\hat{\Phi}_{N-2}$, and the continuation values along the sample paths, $\hat{\Phi}_{N-2}\left(X_{T_{N-2}}^{(m)}\right)$. We then repeat the same procedure until T_0.

3.2. Learning Networks and Option Pricing

For learning neural networks, we generate Monte Carlo sample paths using the SDEs in Section 3.2 based on a similar idea to that of the ordinary LSMC method introduced by Longstaff and Schwartz (2001). Herein, we apply nonlinear functions of the MLPs instead of polynomial functions for the basis of the continuation value functions. Additionally, we introduce techniques such as early stopping to improve the fitted continuation functions and avoid possible overfitting or biases in learning and pricing. We also introduce the resampling procedure to avoid a possible bias caused by using the same random samples between learning and valuation, and we regenerate Monte Carlo sample paths for the valuation of Bermudan option prices (see Appendix A for pricing details).

Herein, we summarize a learning procedure, as described in Algorithm 1 below, where the underlying price and the volatility vector are denoted by $S_t := [S_{1,t}, S_{2,t}, \ldots, S_{l,t}]^\top$ and $\sigma_t := [\sigma_{1,t}, \sigma_{2,t}, \ldots, \sigma_{l,t}]^\top$, while g denotes a payoff function of S_t. Given simulation the sample paths generated by the multi-asset stochastic volatility models in (4), we provide a Bermudan option pricing procedure involving the algorithm for estimating the continuation values using neural networks. This algorithm operates to find MLP Φ as a continuation value function satisfying Equation (8) in the previous section and gives the Bermudan option price.

Algorithm 1. Bermudan option pricing with learning networks.

```
Require: Initiate paths $S_t^{(j)}, \sigma_t^{(j)}, t = T_0, T_1, \cdots, T_N, j = 1, 2, \cdots, M$
Let $p$ be the patience and $Max_{iter}$ be the maximum number of epochs
Put $V^{(j)} \leftarrow g\left(S_{T_N}^{(j)}\right)$ for all $j$
for $t$ from $T_{N-1}$ to $T_1$ do
    Let $X^{(j)} \leftarrow S_t^{(j)}, \sigma_t^{(j)}$ and $V^{(j)} \leftarrow e^{-r\delta t} \cdot V^{(j)}$ for all $j$
    if $t$ on exercisable periods then
        Perform learning on $X$ to obtain network $\Phi_t$ with $Z$ to be $V$
        $i \leftarrow 0$
        $k \leftarrow 0$
        while $i < Max_{iter}$ do
            Train $\Phi_t$ on $X$ and $V$
            if improved then
                $k \leftarrow 0$
            else
                $k \leftarrow k + 1$
            end if
            if $k == p$ then
                Break
            end if
            $i \leftarrow i + 1$
        end while
        Calculate the continuation value $\Phi_t\left(X^{(j)}\right)$ for all $j$
        for $j$ from 1 to $M$ do
            if $g\left(S_t^{(j)}\right) > \Phi_t\left(X^{(j)}\right)$ then
                $V^{(j)} \leftarrow g\left(S_t^{(j)}\right)$
            end if
        end for
    end if
end for
return mean of $e^{-r\delta t} \cdot V$
```

Note This study does not use the selection technique, which performs regression using only the in-the-money paths proposed by Longstaff and Schwartz (2001), for the purpose of constructing a versatile algorithm.

It is noted that one cycle of training with the complete training data is known as an epoch and is repeated for learning purposes for each continuation value function in Algorithm 1. In general, the larger the number of epochs, the better learning of the training data. However, a large number of epochs usually requires a long computational time, even with large computer resources, and sometimes leads to overfitting of the training data. To prevent such situations, we introduce an early stopping rule for the learning procedure given a specified integer p in Algorithm 1. Under the early stopping rule, the objective function (3) is monitored for improvement, and the number of iterations (i.e., the number of epochs) without improvement (compared with the previous epoch), denoted by k, are counted. If this number reaches p, the iteration stops and the learning procedure of the continuation value function terminates; otherwise, the iteration continues as long as the iteration index i is less than Max_{iter}, where Max_{iter} is the maximum number of epochs specified at the beginning of Algorithm 1. Note that the introduction of the early stopping rule not only decreases the computational time but also prevents overfitting/underfitting for the MLP.

Based on the network configuration of the neural networks, the computational complexity of Algorithm 1 is given by the number of iterations for parameter estimation of the MLP. This number of iterations depends on the maximum number of epochs and the number of exercisable dates, Max_{iter} and $N - 1$. Once these values are specified, the maximum number of iterations is $Max_{iter} \times (N - 1)$, which is the total number of epochs applied in Algorithm 1. In addition, the computational complexity of each epoch in the MLP depends on the network configuration (see Serpen and Gao 2014).

To price the Bermudan commodity options using the continuation functions estimated in Algorithm 1, we regenerate different sample paths from those used in the learning procedure for computing the continuation values and Bermudan commodity option prices, given the neural networks in Algorithm 1, i.e., we separate the learning and the valuation procedures, and Algorithm 1 may be applied without learning (i.e., given the estimated neural networks) for the valuation procedure. The merit of this resampling is that it avoids a price bias, which results from overfitting using the same sample paths. Accordingly, this study adopts the following procedure:

1. Generate the sample paths of the underlying assets for the MLP learning of Algorithm 1.
2. Find the MLP network parameters via learning in Algorithm 1 using the sample paths in Step 1.
3. Given the estimated neural networks, regenerate a different set of sample paths and apply Algorithm 1 (without learning) to compute the continuation values and the initial prices of the Bermudan options.
4. Repeat Step 2 and calculate statistical values such as the mean and the standard deviation of the Bermudan option prices.

In the above, it is key that learning and pricing (i.e., valuation) utilize the different simulation sample paths set in Steps 1 and 3. Figure 2 shows a flowchart of the entire procedure for learning and valuation.

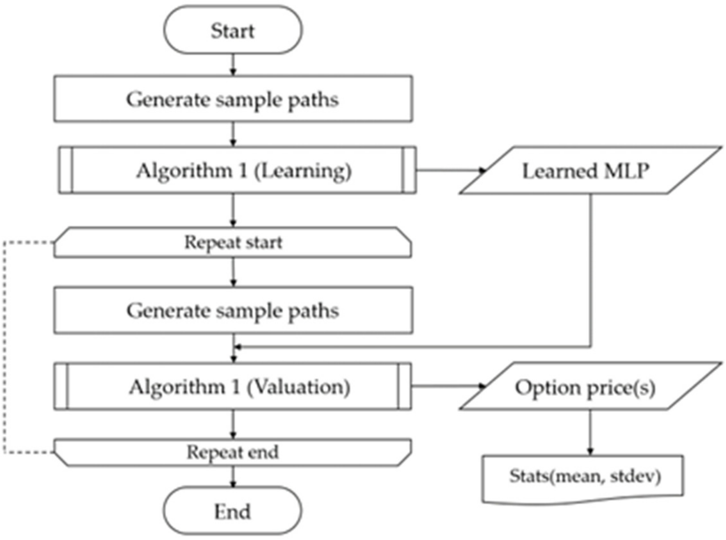

Figure 2. Flowchart of the entire procedure for learning and valuation of Bermudan option pricing.

4. Numerical Experiments

The objective of this section is to execute numerical experiments based on the learning and valuation procedure explained in the previous section and make comparisons of the Bermudan option pricing between the MLP and the benchmark polynomial regression (i.e., the standard (naïve) LSMC method by Longstaff and Schwartz 2001).

4.1. Problem Setting and Preliminary Experiment

Herein, we consider Bermudan commodity options with early-exercisable dates in discretized periods until maturity T, i.e., $0 = T_0 < T_1 < \ldots < T_N = T$, the payoffs of which are given by $g(S_t)$ when exercised. We define several settings for different dimensions of Bermudan options, d_l (i.e., the number of state variables in (5)), exercisable dates, and payoff functions. We also introduce a constant volatility model as a one-dimensional problem to perform a preliminary experiment.

For the exercisable dates of the Bermudan commodity options, we consider two cases as depicted in Figure 3. One is a two-period problem, wherein the Bermudan commodity option is issued at time T_0 and can be exercised at T_1 and maturity T_2. The other is a case with multiple exercisable dates, wherein we choose ten exercisable timings before maturity. In both cases, the options can be exercised after a half-year period to compare the continuation value surfaces at time T_1 between different methodologies, and the values of the options are evaluated at the initial time period, T_0.

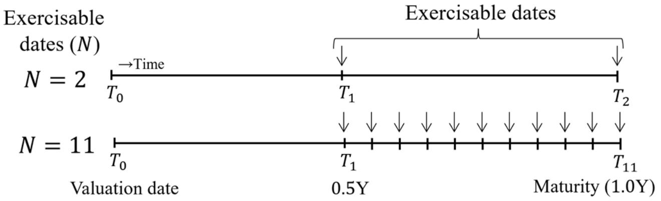

Figure 3. Bermudan option schedules with exercisable dates of 2 and 11 times.

Moreover, the payoff functions for Bermudan basket put options are given as

$$g(S_t) = \max\left(K - \frac{1}{l}\sum_i S_{i,t}, 0\right). \tag{9}$$

Then, an upper limit price D is introduced to the payoff function as

$$g(S_t) = \min\left(\max\left(K - \frac{1}{l}\sum_i S_{i,t}, 0\right), D\right) \tag{10}$$

for Bermudan capped put options. Note that the upper limit, D, provides an additional complexity of the payoff functions.

The parameters of the underlying assets and neural networks are set as shown in Tables 1 and 2 below.

4.2. Low-Dimensional Case: Single-Asset Bermudan Options with Stochastic Volatility and Constant Volatility

We begin with the simplest valuation based on Schwartz's (1997) single-asset and constant volatility model in, compared with the standard LSMC method using polynomial regression and the finite difference method (FDM) detailed by Tavella and Randall (2000). In the MLP, we applied Algorithm 1 for learning neural networks with Monte Carlo simulations and then repeated the valuation procedure (using Algorithm 1 without learning) 100 times. The MLP in this experiment contains two hidden layers with sixty-four neurons of sigmoid activate functions. In the standard LSMC method, we use a quintic function for one-dimensional problems (i.e., $d_l = 1$) and a multi-dimensional quadratic function for two or more higher-dimensional problems (i.e., $d_l \geq 2$) and apply the same procedure (i.e., the learning and valuation procedure). The FDM is based on the Crank–Nicolson scheme with discretized 2000/200/50 grids in the time/asset/volatility directions.

Table 1. Parameters of the underlying asset model with constant and stochastic volatility.

Parameter	Constant Vol. (d_l=1)	Stochastic Vol. ($d_l \geq 2$) [1]
Spot rate (S_0)	100.0	
Strike rate (K)	105.0	
Capped rate (D)	10.0	
Time to maturity (T) [years]	1.0	
Risk-free interest rate (r) [%]	6.0	
Initial volatility ($\sigma_{i,0}$) [%]	30.0	
Long-run mean (μ)	4.8	
Kappa of asset (κ_{S_i})	0.3	
Long-term volatility ($\sqrt{\vartheta_i}$) [%]	-	30.0
Correlation ($\rho_{S_i,\ v_i}$)	-	-0.1
Corr. among assets ($\rho_{S_i,\ S_j}$)	-	0.7 (1.0 $if\ i = j$)
Corr. assets and vol. ($\rho_{S_i,\ v_j}$)	-	-0.07
Corr. among vols. ($\rho_{v_i,\, v_j}$)	-	0.007 (1.0 $if\ i = j$)
Kappa of vol. (κ_{v_i})	-	1.5
Vol. of vol. (ξ_i)	-	0.2
Num. of paths (learning)	100,000	
Num. of paths (valuation)	10,000	
Sim. path timesteps (per yr.)	20	

[1] We applied Euler's method as a discretized method.

Table 2. Neural networks' learning parameters.

Learning Parameters	Value
Num. of sim. paths (M)	100,000
Batch size	4096
Max. num. of epochs (Max_{iter})	200
Train paths percentage	80%
Evaluation paths percentage	20%
Optimizer	Adam [1]

[1] Learning optimizer Adam (Kingma and Ba 2014) hyperparameters are set to 0.01 for the learning rate, 0.9 for beta1, 0.999 for beta2, and 1×10^{-7} for epsilon; training is completed when the loss does not improve even after 20 epochs, as early stopping. Randomized 20% of input paths are used in evaluations to avoid over-learning.

Table 3 compares the means and standard deviations of the option prices obtained with the MLP and the standard LSMC methods. Considering the option price of the FDM as a proxy for the value of the Bermudan commodity option price, we see that both the MLP and the standard LSMC method almost achieve the Bermudan option price value, i.e., the gap between the three prices is sufficiently small for this one-dimensional problem. We implemented Algorithm 1 using the MLP and the standard LSMC on Python, using a machine learning package based on TensorFlow (Abadi et al. 2015) and the PolynomialFeatures toolbox of the scikit-learn library (Pedregosa et al. 2011). All our numerical experiments were run using Google Colaboratory (Google 2022) with 36 GB of RAM and a dual-core CPU of 2.3 GHz.

Table 3. Bermudan (capped) put option prices in single-asset constant volatility ($d_l = 1$).

Bermudan Put Option (d_n=1)				Bermudan Capped Put Option (d_n=1)			
	# of Ex.	**N = 2**	**N = 11**		**# of Ex.**	**N = 2**	**N = 11**
LSMC price	Mean	11.474	11.786	LSMC price	Mean	5.731	6.244
	(St. dev.)	(0.071)	(0.069)		(St. dev.)	(0.027)	(0.029)
MLP price	Mean	11.471	11.812	MLP price	Mean	5.729	6.320
	(St. dev.)	(0.069)	(0.061)		(St. dev.)	(0.027)	(0.031)
FDM price		11.415	11.808	FDM price		5.752	6.350

note FDM = finite difference method; LSMC = least-squares Monte Carlo; MLP = multi-layer perceptron.

When considering a single-asset stochastic model, the corresponding Bermudan commodity option problem becomes two-dimensional, i.e., $d_l = 2$, we can observe a slight difference between the MLP and the LSMC methods, compared with the approximate solution of the FDM, as shown in Table 4. First, we see that there is no significant difference between the 3 cases for the Bermudan put option with exercisable dates $N = 2$. However, the gap between the LSMC and FDM prices becomes slightly wider, compared with that between the MLP and FDM prices with exercisable dates $N = 11$. For the Bermudan capped put option, there is a slightly larger difference vis-à-vis the FDM price for both the MLP and LSMC prices, whereas a slight improvement was achieved by using the MLP in terms of the gap from the FDM price, as illustrated in the box plots in Figure 4.

Table 4. Bermudan (capped) put option prices in single-asset stochastic volatility ($d_l = 2$).

Bermudan Put Option (d_l=2)				**Bermudan Capped Put Option (d_l=2)**			
	# of Ex.	**N = 2**	**N = 11**		**# of Ex.**	**N = 2**	**N = 11**
LSMC price	Mean	11.114	11.170	LSMC price	Mean	5.489	5.985
	(St. dev.)	(0.139)	(0.137)		(St. dev.)	(0.047)	(0.043)
MLP price	Mean	11.113	11.417	MLP price	Mean	5.504	6.015
	(St. dev.)	(0.140)	(0.133)		(St. dev.)	(0.047)	(0.045)
FDM price		11.090	11.460	FDM price		5.541	6.070

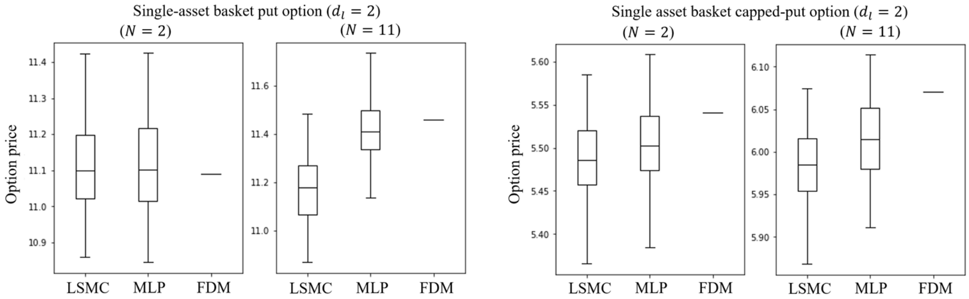

Figure 4. The box plot of option prices with a single asset with stochastic volatility. It has two-dimensional state variables. The conditions were put and capped put payoff, as well as changing exercisable dates N via the method. In the box plot, the center of the box in the first and third quartiles is a median (line), while the beards are the maximum and the minimum values.

4.3. Higher-Dimensional Case: Multi-Asset Bermudan Options with Stochastic Volatility

In the case of higher-dimensional Bermudan commodity options with multi-asset stochastic volatility (e.g., two asset problems with $d_l = 4$), it is difficult (or unrealistic) to obtain an approximate Bermudan option price with high accuracy using the FDM. Thus, we compare option prices obtained with the MLP with the benchmark polynomials (i.e., the standard LSMC method) only, where we set $d_l = 10$, i.e., five-asset stochastic volatility, in the numerical experiments. Then, we will discuss the source of the difference in view of the continuation value functions for both methods in the next section. Additionally, we will compare the accuracy of estimated continuation values by considering a two-asset problem with $d_l = 4$ and two exercisable dates $N = 2$.

Table 5 shows our numerical results, which compare the mean and standard deviation between the LSMC and the MLP prices for $N = 2$ and $N = 11$, obtained via the learning and valuation procedure described in Section 3. In the case of the Bermudan put option for $N = 2$, we see that there is no significant difference between the two methods, as with the case with a low-dimensional problem, $d_l = 2$. However, the gap between the two increased for the Bermudan capped put options and the case with $N = 11$, as shown in the box plots in Figure 5. In other words, we see that the differences between the MLP and the LSMC are emphasized by introducing additional complexity to the payoff function or by increasing exercisable dates.

Table 5. Bermudan (capped) put option prices in the five-asset ($d_l = 10$) stochastic volatility model.

Bermudan Put Option (d_l=10)				**Bermudan Capped Put Option (d_l=10)**			
	# of Ex.	**N = 2**	**N = 11**		**# of Ex.**	**N = 2**	**N = 11**
LSMC price	Mean	9.712	9.803	LSMC price	Mean	5.406	5.884
	(St. dev.)	(0.108)	(0.105)		(St. dev.)	(0.043)	(0.039)
MLP price	Mean	9.712	9.888	MLP price	Mean	5.414	5.904
	(St. dev.)	(0.103)	(0.098)		(St. dev.)	(0.043)	(0.040)

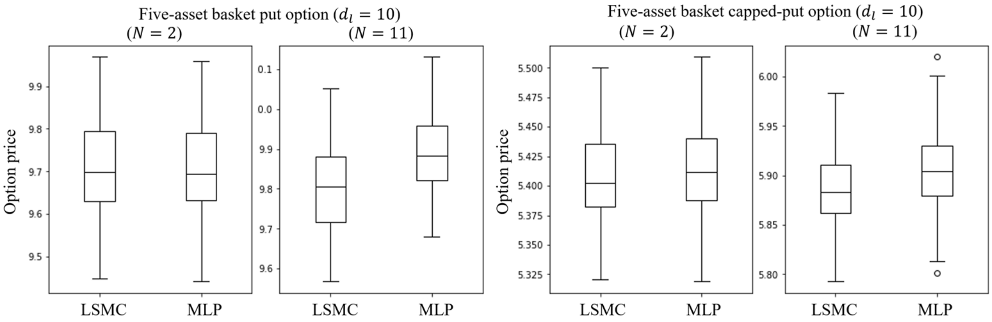

Figure 5. Box plot of option prices' distribution with five-asset stochastic volatility. It has ten-dimensional state variables. The conditions were put and capped-put payoff, as well as changing exercisable dates N via the method. In the box plot, the circles are outliers.

Additionally, we demonstrated the same numerical experiments above but increased the number of assets in the stochastic volatility model to 10 and 20 assets, respectively (i.e., $d_l = 20$ and $d_l = 40$) and compared the estimated Bermudan commodity option prices between the MLP and the LSMC. To observe the effects of increasing the number of exercisable dates more clearly, we added $N = 6$ between $N = 2$ and $N = 11$ to obtain the estimation values shown in Tables 6 and 7 (corresponding to the cases of $d_l = 20$ and $d_l = 40$, respectively). Table 6 shows the means, the standard deviations, and the gaps between the estimated prices for the Bermudan put options and Bermudan capped put options with $d_l = 20$, and Table 7 shows those with $d_l = 40$.

Table 6. Bermudan (capped) put option prices in the 10-asset ($d_l = 20$) stochastic volatility model.

Bermudan Put Option (d_l=20)					**Bermudan Capped Put Option (d_l=20)**				
	# of Ex.	**N = 2**	**N = 6**	**N = 11**		**# of Ex.**	**N = 2**	**N = 6**	**N = 11**
LSMC price	Mean	9.539	9.595	9.571	LSMC price	Mean	5.359	5.747	5.848
	(St. dev.)	(0.120)	(0.098)	(0.097)		(St. dev.)	(0.036)	(0.042)	(0.041)
MLP price	Mean	9.548	9.681	9.684	MLP price	mean	5.376	5.765	5.851
	(St. dev.)	(0.112)	(0.091)	(0.095)		(St. dev.)	(0.035)	(0.041)	(0.041)
Difference (MLP—LSMC)	Mean	0.009	0.087	0.113	Difference (MLP—LSMC)	Mean	0.016	0.018	0.003

Table 7. Bermudan (capped) put option prices in the 20-asset ($d_l = 40$) stochastic volatility model.

Bermudan Put Option (d_l=40)					**Bermudan Capped Put Option (d_l=40)**				
	# of Ex.	**N = 2**	**N = 6**	**N = 11**		**# of Ex.**	**N = 2**	**N = 6**	**N = 11**
LSMC price	Mean	9.338	9.4855	9.453	LSMC price	Mean	5.362	5.728	5.828
	(St. dev.)	(0.112)	(0.111)	(0.111)		(St. dev.)	(0.048)	(0.042)	(0.037)
MLP price	Mean	9.369	9.596	9.600	MLP price	Mean	5.382	5.750	5.833
	(St. dev.)	(0.111)	(0.105)	(0.104)		(St. dev.)	(0.048)	(0.040)	(0.037)
Difference (MLP—LSMC)	Mean	0.031	0.111	0.147	Difference (MLP—LSMC)	Mean	0.020	0.022	0.005

Similar to the previous cases, the gap between the MLP and LMSC increases for the Bermudan put prices given a larger number of exercisable dates but decreases for the Bermudan capped put options when $N = 11$ for both $d_l = 20$ and $d_l = 40$. It is possible that the choice of regression function has a weaker effect for higher dimensional Bermudan capped put options with a larger number of exercisable dates, i.e., the continuation value functions of the capped put options may become flatter or smoother when the number of exercisable dates increases and can be fitted with less sophisticated functions. This phenomenon should be investigated in more detail in a future study.

4.4. Comparison of Continuation Value Surfaces

In the previous subsection, we observed that there are some differences between the MLP and the LSMC regarding the estimated prices and that these differences were more notable for Bermudan capped put options and/or increased exercisable dates. Herein, we discuss the possible reason for this price difference by visualizing the continuation value surfaces for both the MLP and the LSMC methods. For visualization purposes, we consider single-asset Bermudan capped put options with stochastic volatility and constant volatility, i.e., the low-dimensional cases with $d_l = 1$ and $d_l = 2$ introduced in Section 4.2.

The left-hand plot of Figure 6 illustrates the continuation value function estimated at T_1 in the one-dimensional Bermudan capped-put option problem (corresponding to the single-asset constant volatility model) using the LSMC, whereas the right-hand plot shows the problem using the MLP. We first observe that the continuation value function of the MLP monotonically decreases with the underlying price, whereas the one obtained from the LSMC method is a nonmonotonic function. Since the payoff function of the Bermudan capped put option is piecewise linear and it monotonically decreases with the underlying price, the monotonicity of the continuation value function is more consistent with the payoff structure of the Bermudan capped-put option. In this sense, we see that the continuation value function of the MLP reflects the monotonic property more appropriately than that of the LSMC method in the one-dimensional single-asset problem.

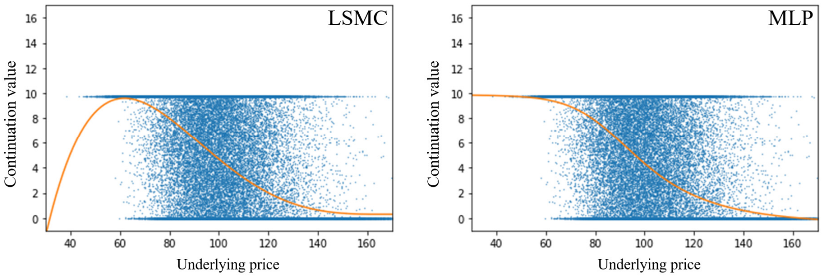

Figure 6. Continuation value functions (red lines) for the one-dimensional problem (with constant volatility) in the Bermudan capped option with exercisable dates $N = 2$. The blue dots represent the generated sample points of Bermudan option payoffs at expiration T_2.

For the two-dimensional case with single-asset stochastic volatility for Bermudan capped put options, the continuation value functions have three-dimensional surfaces, as shown in Figure 7, wherein the left-hand and right-hand plots depict the continuation values with respect to volatility and underlying asset price directions for the LSMC method and the MLP, respectively. As in the one-dimensional problem, the payoff function is piecewise linear with respect to the underlying asset price direction and is flat when the underlying asset price exceeds the strike price $K = 105$ or is less than a certain value related to the capped rate $D = 10$. Since continuation value functions are supposed to approximate the payoff function at the maturity T_2, given the information up to time T_1, their surfaces are expected to have similar shapes, i.e., continuation values are approximately zero or close to the capped rate for larger or smaller values of the underlyings, respectively. In view of this payoff structure for the Bermudan capped put options, the continuation value surface of the MLP seems to approximate the payoff function more accurately than that of the LSMC.

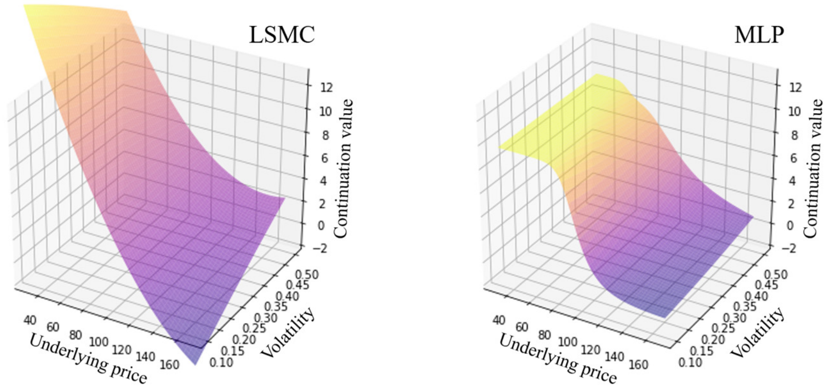

Figure 7. Continuation value surfaces at T_1 of the two-dimensional problem (single-asset stochastic volatility) for Bermudan capped put options.

Remark 1. *In general, the visualization of nonparametric methods provides an intuitive interpretation of the estimated functions. We have observed that "the continuation value function of the MLP monotonically decreases with the underlying price, whereas the one obtained from the LSMC method is a non-monotonic function," and that "since the payoff function of the Bermudan capped-put option is piecewise linear and it monotonically decreases with the underlying price, the monotonicity of the continuation value function is more consistent with the payoff structure of the Bermudan capped-put option," as stated earlier in this section. Such a visualization helps in understanding the valuation structure for the applied method in the middle of the process for Bermudan option pricing, but the effect of the approximation error may be weakened in the total procedure. However, we should be able to understand the approximation errors of estimated continuation value functions intuitively in the middle of the process from such a visualization.*

4.5. Comparison of Accuracy in Continuation Values

In the previous subsection, we observed that the continuation value surfaces of the MLP may approximate the payoff functions more accurately than those of the LSMC by visualizing the continuation value surfaces. To further investigate the estimated continuation value surfaces in higher-dimension problems, we next measure the accuracy of the continuation values using a four-dimensional problem of the Bermudan capped put basket option with two exercisable dates (i.e., $d_I = 4$ and $N = 2$) for both the MLP and the LSMC method.

Consider a problem of estimating the continuation values at T_1, as shown in Figure 8. Since Bermudan options with exercisable dates $N = 2$ become simple European options if not exercised at T_1, the continuation values at T_1 may be estimated via European option prices expiring at T_2, given the state variables at T_1, i.e., $X_{T_1} = \left[S_{1,T_1}, \sigma_{1,T_1},\ S_{2,T_1}, \sigma_{2,T_2}\right]^\top$. Therefore, we can calculate the accuracy of the estimated continuation values at T_1 by measuring the differences between the estimated continuation values and the European option prices at T_1 by specifying different state variables as input values of the European options.

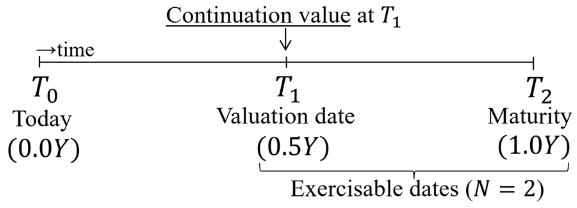

Figure 8. Continuation value at T_1 with two exercisable dates.

We first solved the Bermudan option problem with $d_l = 4$ and $N = 2$ using the same parameter settings as those in the previous subsections by applying the MLP and the LSMC methods; subsequently, we calculated the continuation values at T_1, given the state variables specified in Table 8. Note that Table 8 defines the discretized domain of the state variables, wherein each state variable is discretized in the interval between the minimum and maximum values so that the number of grid points in 1 dimension becomes $2^4 + 1$. Then, the total number of grid points is given by 83,521 ($= 17^4$). Similarly, we applied the standard Monte Carlo simulation to compute the European option price on each grid and repeated this procedure 83,521 times to estimate the surface of the European option prices. This surface of the European option prices may be considered to provide an approximation of the theoretical continuation value (see notes in Table 8), and we can measure the accuracy of the continuation values using the difference between the estimated continuation values with the MLP and the LSMC methods and the simulation-based (theoretical) surface.

Table 8. State variables at T_1. Each state variable is equally discretized into the number of points from minimum up to maximum value.

Variable	Minimum	Maximum	# of Points [1]	Interval
S_{1,T_1}	25.0	175.0	$2^4 + 1$	9.375
σ_{1,T_1}	0.05	0.55	$2^4 + 1$	0.03125
S_{2,T_1}	25.0	175.0	$2^4 + 1$	9.375
σ_{2,T_1}	0.05	0.55	$2^4 + 1$	0.03125

[1] The number of grid points with state variables at T_1 are 83,521 ($= 17^4$) in total. The number of sample paths generated for evaluating each European option in the standard Monte Carlo simulation is 10,000. The average values of estimated European prices and standard errors are 4.997 and 0.027, respectively, indicating that the 95% confidence interval is given by 4.997 ± 0.053 on average for the Monte Carlo simulations.

Additionally, we can change the number of hidden layers/neurons and the type of activation function in the MLP to verify their effects on its accuracy. In this study, we evaluate the size of accuracy in terms of the following normalized root-mean-squared error (NRMSE) for each methodology:

$$NRMSE = \frac{1}{\sqrt{I}} \frac{\sqrt{\sum_{i=1}^{I} (p_i - \hat{p}_i)^2}}{\hat{p}_{max} - \hat{p}_{min}}, \tag{11}$$

where I is the total number of grid points (i.e., $I = 83,521$), and p_i and $\hat{p}_i$ are the i^{th}-continuation value and the corresponding European option price on the same grid point. In Equation (11), the root-mean-squared error is normalized by the difference between $\hat{p}_{min}$ and $\hat{p}_{max}$, which are the minimum and maximum values of European option prices over the entire grid.

We computed the NRMSEs for different settings of neural networks in the case of the MLP, as shown in Table 9, wherein we changed the number of hidden layers/neurons and applied two types of activation functions, i.e., the ReLU and sigmoid functions. Note that the NRMSE with the LSMC method is also computed, as shown in the bottom row of the table, while Figure 9 compares the same NRMSE with respect to a different number of neurons for 16, 32, and 128 using bar graphs. In Table 9, we first observe that the MLP almost always provided better accuracy in terms of NRMSEs compared with the LSMC. Second, when comparing the types of activation functions, the MLP with the sigmoid function was always better than the MLP with the ReLU function for estimating continuation value surfaces when the number of hidden layers/neurons was fixed. This may be explained by the smoothness of the sigmoid function; the continuation value functions are expected to be smooth with respect to the state variables and can be fitted via smooth functions (e.g., the sigmoid function) better than non-smooth functions, such as the ReLU function. Third, an increase in the number of hidden layers is effective for a few hidden layers but does not necessarily improve the NRMSE when the number of hidden layers is three or larger for both MLPs with the ReLU and sigmoid functions. However, in any case, we obtained better NRMSEs by using the MLP with the sigmoid function.

Table 9. NRMSE comparisons with continuation values at T_1 by network settings of the MLP. It differs by activation functions and the number of hidden layers/neurons.

NRMSE										
# of Neurons	**16**		**32**		**64**		**128**		**256**	
# of Hidden Layers	**ReLU**	**Sigmoid**	**ReLU**	**Sigmoid**	**ReLU**	**Sigmoid**	**ReLU**	**Sigmoid**	**ReLU**	**Sigmoid**
1	0.266	0.075	0.262	0.099	0.198	0.160	0.244	0.162	0.160	0.211
2	0.176	0.101	0.145	0.078	0.161	0.081	0.154	0.073	0.311	0.100
3	0.215	0.083	0.246	0.072	0.126	0.097	0.157	0.083	0.271	0.085
4	0.128	0.115	0.225	0.098	0.172	0.079	0.124	0.059	0.184	0.082
5	0.137	0.097	0.132	0.100	0.169	0.090	0.132	0.127	0.154	0.076
LSMC	0.244									

$^{\text{note}}$ NRMSE = normalized root-mean-squared error.

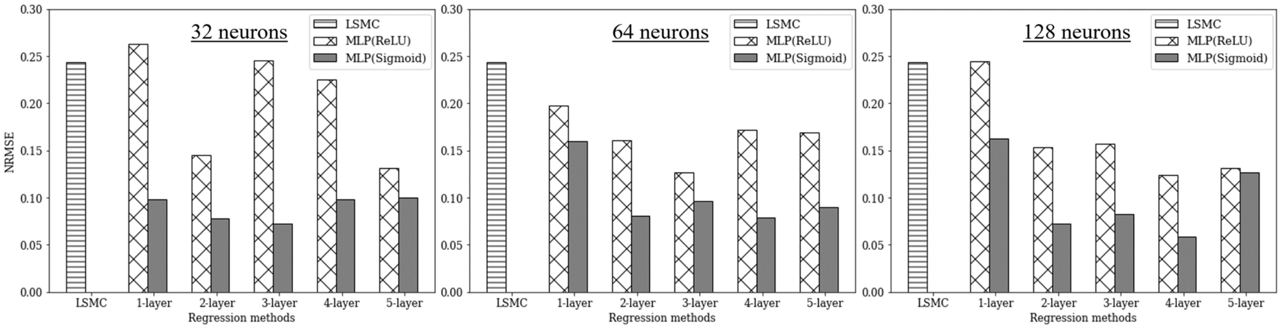

Figure 9. NRMSE comparison with continuation values at T_1. MLP differences in a number of hidden layers, neurons, and selected activation functions.

5. Discussion of Robustness and Computational Costs

In this section, we discuss the robustness and computational costs of our experiment for the pricing algorithm of multi-asset Bermudan commodity options using the MLP.

First, we discuss the learning adaptability of the MLP applied to our multi-asset Bermudan commodity option problems with stochastic volatility. Figure 10 depicts changes in the mean and standard deviation of learning rates with respect to the number of epochs in the MLP. In this figure, we see that training and validation losses remain close, which indicates no overfitting. Furthermore, the learning rate decreases rapidly until the number of epochs is 10 and stays at sufficiently good levels thereafter.

Next, we estimated the computational costs of the learning and valuation procedure when the problem dimension is increased. Table 10 compares the computational costs with respect to the dimensions of $d_l = 2,\ 4,\ 8,\ 16,\ 32,\ 64$, wherein the same numerical experiment as that of the previous section was repeated 100 times and computed the mean and standard deviation of the computational time for each algorithm. Furthermore, the average numbers of epochs in learning are also computed in the MLP. Although the LSMC generally performs much better in terms of computational costs when the dimension is particularly low, its computational time grows exponentially in tandem with the size of the dimension. This is because when using a polynomial regression in the LSMC, the number of terms in the polynomial function increases combinatorially with the number of variables, even though its maximum order is fixed.

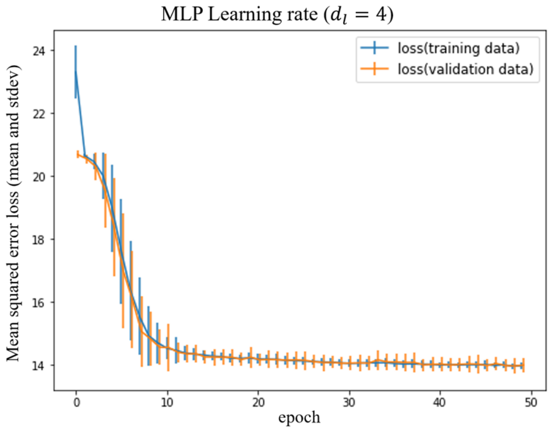

Figure 10. Learning rate by epochs. The learning of the MLP was repeated 100 times and computed the mean (line) and standard deviation (bar) of learning losses in the Bermudan capped put option with the exercise dates $N = 2$ and two underlying assets with stochastic volatility (i.e., $d_n = 4$).

Table 10. Computational cost comparisons. Each algorithm's computational time periods (seconds) and their statistics by dimensions were calculated 100 times in the same option condition, as shown in Figure 10. In the MLP, the average numbers of epochs in learning are also listed.

	LSMC [Sec.]				**MLP [Sec.]** [2]				
Dimensions	**Learning** [1]		**Valuation**		**Learning**			**Valuation**	
(d_l)	**Mean**	**(St. Dev.)**	**Mean**	**(St. Dev.)**	**Mean**	**(St. Dev.)**	**# of Epochs**	**Mean**	**(St. Dev.)**
2	0.025	(0.002)	0.003	(0.003)	20.038	(3.599)	186.19	0.006	(0.002)
4	0.076	(0.019)	0.003	(0.001)	17.637	(4.403)	156.56	0.006	(0.002)
8	0.257	(0.064)	0.005	(0.001)	17.382	(5.089)	157.43	0.006	(0.001)
16	0.861	(0.083)	0.020	(0.004)	9.773	(7.707)	81.99	0.007	(0.002)
32	3.708	(0.334)	0.068	(0.020)	5.114	(2.829)	38.53	0.008	(0.002)
64	19.887	(2.842)	0.210	(0.018)	5.367	(1.468)	36.58	0.009	(0.002)

[1] LSMC with a multi-dimensional quadratic function. [2] MLP with 2 hidden layers with 64 neurons of sigmoid activation functions.

In contrast with the LSMC, the computational cost in the MLP is mostly unaffected by the size of the dimension but is directly proportional to the average number of epochs in learning. The computational cost of learning depends on how often the networks are updated during training, but the computation cost per one cycle of training data (i.e., epoch) remains the same when the size of the network is fixed. In the numerical experiments, we applied two hidden layers with sixty-four neurons using sigmoid activation functions, whereby the computational cost per epoch remained almost the same regardless of dimensions; the computational time in learning is determined by the total number of epochs. Although the computational cost per epoch slightly increases as the number of features in the input layer increases by dimension, the average computational time decreases even for large dimensions with the reduction in the average number of epochs due to the early stopping rule. This is the benefit of introducing the early stopping rule in Algorithm 1, which is particularly effective for higher-dimensional problems to avoid unnecessarily increasing training iterations (and overfitting). Note that the average computational time for both learning and valuation became smaller for the MLP than the LSMC when $d_l = 64$.

6. Conclusions

In this study, we detailed the use of a neural network for pricing multi-asset Bermudan option problems with stochastic volatility in commodity markets and illustrated its effectiveness. First, we employed the MLP to estimate continuation values in the multi-dimensional Bermudan commodity option problem, whereby we formulated the multi-asset stochastic volatility model and generated Monte Carlo simulation sample paths for learning continuation value functions using the MLP. Then, in the applied algorithm, we introduced early stopping into the learning of the MLP to avoid unnecessarily increasing training iterations and overfitting. The early stopping rule was activated by counting the number of epochs without improvement compared with the previous epoch. We also introduced a resampling process, and a valuation procedure was applied for the estimated neural networks by generating a different set of simulation sample paths. We executed numerical experiments to evaluate the accuracy of the continuation values and the initial price of the option using different settings of networks, problem dimension, and exercisable dates, whereby two types of payoff functions for Bermudan commodity options were considered, namely Bermudan put options and Bermudan capped put options. From our numerical analysis, we clarified the following observations:

1. No significant difference was observed between the MLP and the standard LSMC method when solving Bermudan put option problems with a few exercisable dates. However, there was a slight difference for the Bermudan capped put options; this difference was emphasized when the number of excisable dates increased. A similar tendency was observed for higher-dimensional cases, but the gap narrowed between the mean values of the two methods for Bermudan capped put options, as shown in our additional numerical experiments.
2. While it turned out that the MLP was not much better than the standard LSMC from the numerical experiments in Section 4.3 for high-dimensional cases, it is meaningful to show how the accuracy and computational time can be achieved using the current computational resources. In addition, we expect that the MLP has the potential to achieve much better accuracy due to its generality and flexibility. Moreover, if computational power is increased, the MLP should become more efficient since computational effort grows slower than that of polynomial regressions in the standard LSMC for higher-dimensional problems, as illustrated in the numerical experiments in Section 5.
3. From the perspective that the continuation value function is expected to approximate the payoff function (given state variables) one step before maturity, the shape of the continuation values from the MLP reflected the structure of payoff functions more accurately than the LSMC method.
4. Based on the fact that the continuation values of Bermudan options one step before maturity can be computed as European option prices, we measured the accuracy of the estimated continuation values and examined the effects of different network configurations in the MLP, changing the number of hidden layers/neurons and the choice of activation functions. We observed that the MLP almost always provided better accuracy in terms of NRMSEs compared with the LSMC; furthermore, when comparing the types of activation functions, the MLP with the sigmoid function was always better than the MLP with the ReLU function for estimating continuation value surfaces. An increase in the number of hidden layers was effective for a few network layers but did not necessarily improve the accuracy when the number of hidden layers was three or larger.
5. We computed the learning rate by epochs to show the learning adaptability of our proposed algorithm using the MLP, which indicated no overfitting and achieved sufficiently good levels of learning rates at approximately 10 epochs. Additionally, we showed that although the LSMC generally performs significantly better in terms of computational costs when the dimension is particularly low, its computational time grows exponentially with the size of the dimension due to the combinatorial characterization with respect to the number of terms in the polynomial functions. Conversely, the computational costs of the MLP were mostly unaffected by the size of the dimension or even decreased for large dimensions due to the introduction of the early stopping rule.

Essentially, the use of neural networks for option pricing has the advantage of recognizing sizable input features and generating flexible output features in a unified framework. Nevertheless, there are some drawbacks: high computational effort and resources are required for learning networks, especially for exotic options, including the Bermudan commodity options considered in this study. Additionally, it is necessary to frequently re-learn the networks in response to market conditions. However, we observed that the neural network approach using the MLP reached an appropriate level of learning rates at around 10 epochs, even in high-dimensional cases, as illustrated in our numerical experiments. Therefore, this approach is expected to reduce learning costs if network configurations are developed appropriately.

Although this study chose a relatively simple structure of multi-layered networks, there are other types of network structures such as a recursive structure and unsupervised learning, as discussed in various fields, including pattern recognition and time-series prediction. In finance, although some examples of recursive neural networks for time-series analyses exist, to the best of our knowledge, their use for option pricing has not been considered sufficiently. Moreover, it is important to use empirical data and demonstrate the practicability and applications for risk management in actual commodity market businesses. Such further investigation is interesting and could be considered potential topics for future study.

Author Contributions: Conceptualization, K.H. and Y.Y.; methodology, K.H. and Y.Y.; software, K.H.; validation, K.H. and Y.Y.; formal analysis, K.H.; investigation, K.H.; resources, K.H. and Y.Y.; data curation, K.H.; writing—original draft preparation, K.H.; writing—review and editing, Y.Y.; visualization, K.H.; supervision, Y.Y.; project administration, Y.Y.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: Grant-in-Aid for Scientific Research (A) 20H00285 and Grant-in-Aid for Challenging Research (Exploratory) 19K22024 from the Japan Society for the Promotion of Science (JSPS).

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by the Grant-in-Aid for Scientific Research (A) 20H00285 and Grant-in-Aid for Challenging Research (Exploratory) 19K22024 from the Japan Society for the Promotion of Science (JSPS).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Recursive Formulation for a Bermudan Option Pricing Problem

Given a complete filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{0\leq t\leq T}, \mathbb{P})$ with a finite time horizon $T > 0$, we assume that a set of underlying assets is modeled using a multifactored process $(X_t)_{0\leq t\leq T}$ adapted to the filtration, $(\mathcal{F}_t)_{0\leq t\leq T}$, and that $\mathbb{P}$ is an associated risk-neutral measure. We consider a Bermudan option with exercise dates $0 = T_0 \leq T_1 < T_2 < \ldots < T_N = T$ and discrete-time payoff process P_{T_n} if exercised at times $(T_n)_{0\leq n\leq N}$, where P_{T_n} is specified as a function of X_{T_n}.

In the Bermudan option, the continuation and exercise values are compared at each exercisable period, while the option is exercised if the exercise value is higher. Therefore, Bermudan option value V_{T_n} is computed using the following recursive equation:

$$\begin{cases} V_{T_N} = P_{T_N} \\ V_{T_{n-1}} = max\left(P_{T_{n-1}}, e^{-r\delta_{T_n}}\mathbb{E}\left[V_{T_n}\middle|\mathcal{F}_{T_{n-1}}\right]\right),\ 1 \leq n \leq N \end{cases}. \tag{A1}$$

In the risk-neutral measure, the conditional expected value of the risk-neutral probability measure $\widetilde{P}$ with the risk-free interest rate r and the interval between T_{n-1} and T_n as δ_{T_n}, based on the filtration $\mathfrak{F}_{T_{n-1}}$ up to time T_{n-1}, indicates the continuation value $U_{T_{n-1}}$ as

$$U_{T_{n-1}} = e^{-r\delta_{T_n}}\widetilde{E}\left[V_{T_n}\middle|\mathfrak{F}_{T_{n-1}}\right]. \tag{A2}$$

Bermudan option value $V_{T_{n-1}}$ at time T_{n-1} is sequentially calculated backward when the continuation value $U_{T_{n-1}}$ is identified. The key is estimating the continuation value as a function that consists of underlying multivariate risk factors. Since underlying assets consist of a multi-dimensional Markov process, the continuation value function can be expressed as the multi-dimensional nonlinear function with the Markov process state variables $X_{T_{n-1}}$,

$$\widetilde{E}\big[V_{T_n}\big|\mathfrak{F}_{T_{n-1}}\big] = \widetilde{E}\big[V_{T_n}\big|X_{T_{n-1}}\big]. \tag{A3}$$

Furthermore, from the definition of the conditional expectation, there is a measurable function $h_{T_{n-1}}$ that satisfies the following equation:

$$h_{T_{n-1}}\big(X_{T_{n-1}}\big) = e^{-r\delta_{T_n}}\widetilde{E}\big[V_{T_n}\big|X_{T_{n-1}}\big]. \tag{A4}$$

For the approximation of a function $h_{T_{n-1}}\big(X_{T_{n-1}}\big)$, we can consider $\Phi_{T_{n-1}}\big(x_{T_{n-1}}\big)$ as the approximation function at time T_{n-1},

$$h_{T_{n-1}}\big(X_{T_{n-1}}\big) \approx \mathit{\Phi}_{T_{n-1}}\big(X_{T_{n-1}}\big). \tag{A5}$$

After that, the price at time $t = 0$ is calculated by following recursive backward procedures using the relationship V_{T_n} and $V_{T_{n-1}}$. At maturity $T_N(=T)$, the continuation value of the Bermudan option is $U_T \equiv 0$. Therefore, the Bermudan option's value at maturity T_N is:

$$V_{T_N} = P_{T_N}. \tag{A6}$$

By (A4), the continuation value $U_{T_{N-1}}$ at time T_{N-1} is expressed as

$$U_{T_{N-1}} = h_{T_{N-1}}\big(X_{T_{N-1}}\big) \approx \mathit{\Phi}_{T_{N-1}}\big(X_{T_{N-1}}\big). \tag{A7}$$

The Bermudan option value at time T_{N-1} is expressed as

$$V_{T_{N-1}} := \max\big(P_{T_{N-1}}, \mathit{\Phi}_{T_{N-1}}\big(X_{T_{N-1}}\big)\big). \tag{A8}$$

We can obtain Bermudan option value V_{T_0} by adapting (A7) and (A8) backward, recursively, each time step to $n = 1$. In multi-asset Bermudan option pricing, Monte Carlo simulations are generally used because other numerical methods become exponentially more difficult in higher-dimensional cases. By simulating a large number of paths, we can use the average of the prices obtained from each path as an estimator of the price as

$$\overline{V_{T_0}} = \frac{1}{M}\sum_{j=1}^{M} V_{T_0}^{(j)}, \tag{A9}$$

where M is the number of simulated paths.

From the above, the prices of the Bermudan options can be obtained by finding the approximate functions of the continuation value functions at each exercisable period.

Appendix B. Least-Squares Monte Carlo Method

The least-squares Monte Carlo (LSMC) method, proposed by Longstaff and Schwartz (2001), is a method of early-exercisable option pricing in which regression calculation uses simulation sample paths. In the LSMC method, a polynomial function of the Markov process state variables is applied to identify the continuation values. The following is an algorithm for Bermudan option pricing using the LSMC method.

Step 0. Generate Monte Carlo sample paths of the underlying asset prices and state variables. We denote the underlying asset prices at time t in the j-th sample path as $S_t^{(j)}$

and the Markov process state variables as $x_t^{(j)}$. Subsequently, we obtain the series of paths as

$$\begin{matrix} S_{T_1}^{(1)}, & S_{T_2}^{(1)}, & \dots, & S_{T_N}^{(1)} \\ S_{T_1}^{(2)}, & S_{T_2}^{(2)}, & \dots, & S_{T_N}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ S_{T_1}^{(M)}, & S_{T_2}^{(M)}, & \dots, & S_{T_N}^{(M)} \end{matrix}, \quad \begin{matrix} x_{T_1}^{(1)}, & x_{T_2}^{(1)}, & \dots, & x_{T_N}^{(1)} \\ x_{T_1}^{(2)}, & x_{T_2}^{(2)}, & \dots, & x_{T_N}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_{T_1}^{(M)}, & x_{T_2}^{(M)}, & \dots, & x_{T_N}^{(M)} \end{matrix}. \tag{A10}$$

Step 1. Calculate the series of Bermudan option values at maturity $T_N(= T)$, as follows:

$$V_{T_N} := \left[g\left(S_{T_N}^{(1)}\right), g\left(S_{T_N}^{(2)}\right), \dots, g\left(S_{T_N}^{(M)}\right)\right]^\top, \tag{A11}$$

where g denotes a payoff function of S_t.

Step 2. Find a polynomial function that approximates the continuation values. Herein, we denote a polynomial function as $\hat{h}_{T_{N-1}}$ and a measurable function in (A4) as $h_{T_{N-1}}$. Then,

$$h_{T_{N-1}}(x_{T_{N-1}}) \approx \hat{h}_{T_{N-1}}(x_{T_{N-1}}). \tag{A12}$$

Additionally, $\hat{h}_{T-1}$ is sought to minimize the following equation:

$$\frac{1}{M}\sum_{i=1}^{M}\left(\hat{h}_{T_{N-1}}(x_{T_{N-1}}) - e^{-r\delta_{T_N}} V_{T_N}\right)^2. \tag{A13}$$

Step 3. Calculate the approximated continuation values. Let $h_{T_{N-1}} \equiv \hat{h}_{T_{N-1}}$ and set the series of approximated continuation values at T_{N-1} as

$$\left[\hat{h}_{T_{N-1}}\left(x_{T_{N-1}}^{(1)}\right), \hat{h}_{T_{N-1}}\left(x_{T_{N-1}}^{(2)}\right), \dots, \dots, \hat{h}_{T_{N-1}}\left(x_{T_{N-1}}^{(M)}\right)\right]^\top. \tag{A14}$$

Step 4. Calculate the exercised values and Bermudan option values. The series of exercise values at T_{N-1} using underlying asset prices (A10) is

$$\left[g\left(S_{T_{N-1}}^{(1)}\right), g\left(S_{T_{N-1}}^{(2)}\right), \dots, g\left(S_{T_{N-1}}^{(M)}\right)\right]^\top. \tag{A15}$$

Then, the series of Bermudan option values $V_{T_{N-1}}$ at time T_{N-1} is obtained as

$$\left[V_{T_{N-1}}\left(S_{T_{N-1}}^{(1)}, x_{T_{N-1}}^{(1)}\right), V_{T_{N-1}}\left(S_{T_{N-1}}^{(2)}, x_{T_{N-1}}^{(2)}\right), \dots, \dots, V_{T_{N-1}}\left(S_{T_{N-1}}^{(M)}, x_{T_{N-1}}^{(M)}\right)\right]^\top, \tag{A16}$$

where

$$V_{T_{N-1}}\left(S_{T_{N-1}}^{(j)}, x_{T_{N-1}}^{(j)}\right) := \max\left(g\left(S_{T_{N-1}}^{(j)}\right), h_{T_{N-1}}\left(x_{T_{N-1}}^{(j)}\right)\right), j = 1, \dots, M, \tag{A17}$$

Step 5. Repeat Step 2 to Step 4 for possible exercise times $T_{N-1}, T_{N-2}, \dots$, until time T_0. A series of Bermudan option values V_{T_0} at time T_0 are obtained by repeating Step 2 to Step 4 backward to time T_0.

Step 6. Calculate the Bermudan option price $\overline{V_{T_0}}$. Equation (A9) gives the Bermudan option price $\overline{V_{T_0}}$ at time T_0.

Using this approach, we calculated the Bermudan option price using the continuation values from the polynomial function at each exercisable time. The main point of the pricing procedure in the LSMC method is that the polynomial function is defined to approximate the continuation values at each exercisable time point.

References

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, and et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. TensorFlow. Available online: https://www.tensorflow.org/ (accessed on 1 December 2022).

Andersson, Kristoffer, and Cornelis W. Oosterlee. 2021. A Deep Learning Approach for Computations of Exposure Profiles for High-Dimensional Bermudan Options. *Applied Mathematics and Computation* 408: 126332. [CrossRef]

Arnold, Vladimir I. 2009. On functions of three variables. Collected Works: Representations of Functions. *Celestial Mechanics and KAM Theory* 1957–1965: 5–8.

Ball, Clifford A., and Antonio Roma. 1994. Stochastic volatility option pricing. *Journal of Financial and Quantitative Analysis* 29: 589–607. [CrossRef]

Ballestra, Luca Vincenzo, and Graziella Pacelli. 2013. Pricing European and American Options with Two Stochastic Factors: A Highly Efficient Radial Basis Function Approach. *Journal of Economic Dynamics and Control* 37: 1142–67. [CrossRef]

Becker, Sebastian, Patrick Cheridito, and Arnulf Jentzen. 2020. Pricing and Hedging American-Style Options with Deep Learning. *Journal of Risk and Financial Management* 13: 158. [CrossRef]

Becker, Sebastian, Patrick Cheridito, Arnulf Jentzen, and Timo Welti. 2021. Solving High-Dimensional Optimal Stopping Problems Using Deep Learning. *European Journal of Applied Mathematics* 32: 470–514. [CrossRef]

Chen, Yangang, and Justin W. L. Wan. 2021. Deep Neural Network Framework Based on Backward Stochastic Differential Equations for Pricing and Hedging American Options in High Dimensions. *Quantitative Finance* 21: 45–67. [CrossRef]

Choon, Ong Hong, Leong Chee Hoong, and Tai Sheue Huey. 2008. A functional approximation comparison between neural networks and polynomial regression. *WSEAS Trans. Math* 7: 353–63.

Clark, Iain J. 2014. *Commodity Option Pricing: A Practitioner's Guide*. New York: John Wiley & Sons.

Cybenko, George. 1989. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals and Systems* 2: 303–14. [CrossRef]

Feng, Guiyun, Guangwu Liu, and Lihua Sun. 2013. A Nonparametric Method for Pricing and Hedging American Options. In *2013 Winter Simulations Conference (WSC)*. Piscataway: IEEE, pp. 691–700.

Google. 2022. Frequently Asked Questions. Available online: https://research.google.com/colaboratory/faq.html (accessed on 1 December 2022).

Hahn, Warren J., and James S. Dyer. 2008. Discrete time modeling of mean-reverting stochastic processes for real option valuation. *European Journal of Operational Research* 184: 534–48. [CrossRef]

Haugh, Martin B., and Leonid Kogan. 2004. Pricing American Options: A Duality Approach. *Operations Research* 52: 258–70. [CrossRef]

Heston, Steven L. 1993. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *The Review of Financial Studies* 6: 327–43. [CrossRef]

Kingma, Diederik P., and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv* 1412: 6980.

Kohler, Michael, Adam Krzyżak, and Nebojsa Todorovic. 2010. Pricing of High-Dimensional American Options by Neural Networks. *Mathematical Finance* 20: 383–410. [CrossRef]

Kolmogorov, Andrei Nikolaevich. 1957. On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition. *Doklady Akademii Nauk* 114: 953–56.

Lapeyre, Bernard, and Jérôme Lelong. 2021. Neural Network Regression for Bermudan Option Pricing. *Monte Carlo Methods and Applications* 27: 227–47. [CrossRef]

Li, Nan. 2022. An Iteration Algorithm for American Options Pricing based on Reinforcement Learning. *Symmetry* 14: 1324. [CrossRef]

Liang, Jian, Zhe Xu, and Peter Li. 2021. Deep Learning-Based Least Squares Forward-Backward Stochastic Differential Equation Solver for High-Dimensional Derivative Pricing. *Quantitative Finance* 21: 1309–23. [CrossRef]

Lin, Jingying, and Caio Almeida. 2021. American Option Pricing with Machine Learning: An Extension of the Longstaff-Schwartz Method. *Brazilian Review of Finance* 19: 85–109. [CrossRef]

Liu, Shuaiqiang, Cornelis W. Oosterlee, and Sander M. Bohte. 2019. Pricing Options and Computing Implied Volatilities Using Neural Networks. *Risks* 7: 16. [CrossRef]

Longstaff, Francis A., and Eduardo S. Schwartz. 2001. Valuing American Options by Simulation: A Simple Least-Squares Approach. *The Review of Financial Studies* 14: 113–47. [CrossRef]

Ludkovski, Michael. 2018. Kriging Metamodels and Experimental Design for Bermudan Option Pricing. *Journal of Computational Finance* 22: 37–77. [CrossRef]

Montesdeoca, Luis, and Mahesan Niranjan. 2016. Extending the Feature Set of a Data-Driven Artificial Neural Network Model of Pricing Financial Options. Paper presented at 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, December 6–9; pp. 1–6.

Pedregosa, Fabian, Ga"el Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and et al. 2011. Scikit-Learn: Machine Learning in Python. *The Journal of Machine Learning Research* 12: 2825–30.

Puka, Radosław, Bartosz Łamasz, and Marek Michalski. 2021. Effectiveness of Artificial Neural Networks in Hedging Against WTI Crude Oil Price Risk. *Energies* 14: 3308. [CrossRef]

Ruf, Johannes, and Weiguan Wang. 2020. Neural Networks for Option Pricing and Hedging: A Literature Review. *Journal of Computational Finance* 24: 1–46. [CrossRef]

Salvador, Beatriz, Cornelis W. Oosterlee, and Remco van der Meer. 2020. Financial Option Valuation by Unsupervised Learning with Artificial Neural Networks. *Mathematics* 9: 46. [CrossRef]

Schwartz, Eduardo S. 1997. The Stochastic Behavior of Commodity Prices: Implications for Valuation and Hedging. *The Journal of Finance* 52: 923–73. [CrossRef]

Serpen, Gursel, and Zhenning Gao. 2014. Complexity analysis of multilayer perceptron neural network embedded into a wireless sensor network. *Procedia Computer Science* 36: 192–97. [CrossRef]

Sirignano, Justin, and Konstantinos Spiliopoulos. 2018. Dgm: A Deep Learning Algorithm for Solving Partial Differential Equations. *Journal of Computational Physics* 375: 1339–64. [CrossRef]

Tavella, Domingo, and Curt Randall. 2000. *Pricing Financial Instruments: The Finite Difference Method*. New York: John Wiley & Sons.

Yamada, Yuji. 2012. Properties of Optimal Smooth Functions in Additive Models for Hedging Multivariate Derivatives. *Asia-Pacific Financial Markets* 19: 149–79. [CrossRef]

Yamada, Yuji. 2017. Optimal Hedging of Basket Barrier Options with Additive Models and Its Application to Equity Value Separation Problem. *Asia-Pacific Financial Markets* 24: 1–18. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.