Machine learning solutions to challenges in finance: An application to the pricing of financial products☆

Lirong Gan[a], Huamao Wang[b], Zhaojun Yang[a]

[a]*Department of Finance, Southern University of Science and Technology, Shenzhen, CN.*
[b]*Kent Business School, University of Kent, Canterbury, Kent CT2 7FS, UK.*

Abstract

The recent fast development of machine learning provides new tools to solve challenges in many areas. In finance, average options are popular financial products among corporations, institutional investors, and individual investors for risk management and investment because average options have the advantages of cheap prices and their payoffs are not very sensitive to the changes of the underlying asset prices at the maturity date, avoiding the manipulation of asset prices and option prices. The challenge is that pricing arithmetic average options requires traditional numerical methods with the drawbacks of expensive repetitive computations and non-realistic model assumptions. This paper proposes a machine-learning method to price arith-

☆We are grateful to the two anonymous reviewers for their helpful advice and comments, which significantly improved the paper. We are grateful for the helpful comments of the participants on the First Annual Conference of Quantitative Finance and Insurance Branch of Chinese Society of Optimization, Overall Planning and Economical Mathematics in Hulun Buir, China. The research reported in this paper was supported by Guangdong Planning Office of Philosophy and Social Science for the 13th Five-Year Plan (project No. GD19CYJ23) and Colleges Innovation Project of Guangdong (project No. 2018WTSCX131). The order of the authors is alphabetical in the surnames and the contributions of the authors are the same.

Email addresses: `11930724@mail.sustech.edu.cn` (Lirong Gan), `h.wang@kent.ac.uk` (Huamao Wang), `yangzj@sustech.edu.cn` (Zhaojun Yang)

metic and geometric average options accurately and in particular quickly. The method is model-free and it is verified by empirical applications as well as numerical experiments.
Keywords: Machine learning; Finance applications; Asian options; Model-free asset pricing; Financial technology.

---

1. Introduction

The accelerating development of computer technology and machine learning attracts increasing research interests in the innovative solution to traditional challenges in social sciences. In the areas of finance and risk management, pricing *arithmetic average options* effectively is a challenge in the industry. In this study, we show that the powerful deep machine learning provides a new effective method to solve this challenge in theory and applications.

Options are one category of financial instruments that are largely traded in industry. They are also referred to as one type of financial derivatives since options are based on other underlying financial securities like corporate stocks. The holders of an option pay a premium, i.e., the *option price*, to obtain the right rather than the liability to trade the corresponding underlying assets at an agreed price called the *strike* price at or within a specific *maturity* date. The difference between the strike price and a quantity that depends on the market prices of the underlying asset forms the *payoff* of the option. Simple *European* options have the payoffs that depend on the current market price of the underlying asset at the maturity date. On the contrary, *average options*, which are also called *Asian options*, have the payoffs that depend on the average price of the underlying asset within the maturity date.

Among diverse options, average options are widely used by companies, institutional investors, and individual investors to hedge against risks and to construct investment portfolios due to two advantages of average options, see, e.g., Fusai and Roncoroni (2007) and Kolb and Overdahl (2010). First, average options avoid manipulating the underlying asset prices to affect option payoffs. The payoff of an average option depends on the average price of the underlying asset over a given period, where the average price is a *geometric* or *arithmetic* average of the prices of the underlying asset. Thus, in contrast to European options, the average options' payoffs are not sensitive to the change of the underlying asset prices within the maturity date, and it is therefore not so profitable to manipulate the price of the underlying asset. Second, the prices of average options are relatively cheap compared with other options whose payoffs depend on the market price of the underlying asset at the maturity date. The reason is that the risk of the average asset price is relatively lower than the risk of the asset price at the maturity date.

In addition to the two above documented advantages of Asian options, another reason for their popularity is that there is a large demand for Asian options in the industry. For instance, indexed annuity contracts that are issued by insurance companies often carry liabilities that are equivalent to the issuance of Asian options. Therefore, insurance companies trade Asian options largely to hedge the embedded option risk. To meet the increasing market demand, the Chicago Board Options Exchange (CBOE), one of the world's largest exchange holding companies, introduced the new product, Asian FLEX Index Options, in April 2016. Only in the first week, open interest for such Asian options soared to more than 680 contracts with a notional value of more than \$6 million.[1]

There are two types of Asian options: one is geometric and the other is arithmetic due to the different interpretations of the word “average”. It is straightforward to price geometric average options following a simple explicit expression. However, there is still no closed-form solution available for pricing arithmetic average options because the distribution of the payoff of an arithmetic Asian options is unknown, even though they have been studied for a long time. Usually, arithmetic Asian options are priced by solving a Partial Differential Equation (PDE) numerically (Vecer, 2001) or by Monte Carlo simulation. For more details, please refer to Yan (2018), Yang et al. (2011) among others. The challenge of pricing arithmetic average options with the traditional numerical methods comes from the drawbacks of expensive repetitive computations and simplified models with non-realistic assumptions.

Pricing arithmetic Asian options effectively is a long-standing problem in finance practice given the fact that arithmetic Asian options are much more popular than geometric Asian options. Most actual Asian options in both the exchange markets and over-the-counter (OTC) markets are arithmetic Asian options, as pointed out by Fusai and Roncoroni (2007) and Kolb and Overdahl (2010). For example, the popular Asian FLEX Index Options are arithmetic Asian options in the exchange market. Another type of popular arithmetic Asian options traded in the market are WTI Average Price Op-

[1]See the post on CBOE Blogs at https://www.cboe.com/blogs/options-hub/2016/04/27/first-trades-new-cboe-flex-index-options-asian-style-settlement.

tions based on oil futures and CMEgroup.com reports that their open interest is up to 413,681 contracts on 17th January 2020. These options are favored in thinly traded asset markets like oil markets, where the trading volume is relatively low but individual transactions are particularly large, because in these markets the manipulation of asset prices is possible. In OTC markets, Average Rate Options are wildly traded to hedge against the adverse movements of foreign exchange rates and many of these options are settled on the arithmetic average prices (Levy, 1992). Unfortunately, in sharp contrast to geometric Asian options, there are no effective closed-form solutions for pricing arithmetic Asian options and this is a long-standing unresolved problem.

In this paper, we propose a machine learning method based on deep learning to price arithmetic and geometric average options. This method is a model-free approach for asset pricing. We highlight the effectiveness of the new method by carrying out a comprehensive numerical experiment with computer-generated data. In addition, we verify the new method through an empirical test with real data and the results highlight the effectiveness of the method. Most of the absolute pricing errors are between ± 0.0015. The median of prediction bias is about 0.8% and the 95% bias mean is less than 2%. The mean square error (MSE) is near zero at 10^{-6}. The value of R^2 and the correlation between the real data and the predicted data are almost 1. Furthermore, the trained deep learning model is able to compute 10,000 Asian option prices in less than 1 second, which is much faster than the exact formula method taking 22 seconds for Geometric average options and the time-consuming simulation method taking 100,000 seconds. In brief, both effectiveness and efficiency of the deep learning method are beneficial to practitioners in the industry who usually have to carry out a large number of computations and make prompt decisions.

Our study is related to the recent literature about the applications of machine learning technologies to price financial options. For instance, Halperin (2017) employs a reinforcement Q-Learning method to learn dynamically the optimization of risk-adjusted returns of a portfolio that replicates European options. Ferguson and Green (2018) show that deep learning is capable of pricing a basket option on a basket of stocks accurately and it is a million times faster than traditional models. Cao et al. (2018) utilize neural networks to examine the volatility surface of the S&P 500 index option, which responses distinctively in high and low volatility environments. In sharp contrast to the literature that considers the pricing of options, for which there are closed-form solutions in early studies, the novelty of our study is that we implement a model-free and data-driven deep learning method to price the popular arithmetic Asian options, for which there has been no closed-form solution all the time. Our method is verified by numerical experiments and empirical applications. To the best of our knowledge, there is no paper pricing arithmetic Asian options with model-free pricing method in the framework of deep learning.

For a complete overview of related literature on the application of machine learning to option pricing, we summarize the following recent studies that consider more complicated processes of jumps and stochastic volatilities. Karatas et al. (2019) price vanilla and exotic options by using deep neural networks under diffusion and jump processes that incorporate stochastic volatilities. They test a variety of loss functions and optimization methods to show that deep neural networks exponentially accelerate option pricing. Fu and Hirsa (2019) use a machine learning technique to reduce the error of the quadratic approximation method for pricing American options under the variance gamma model and show that their method is efficient and accurate compared to the classic methods of finite difference and simulation. Jacquier et al. (2019) apply machine learning methods to learn the control variates in the simulation method of pricing European and Asian options in local stochastic volatility models.

Although these recent studies demonstrate the capability of machine learning in solving complicated option pricing models, we focus on the standard model with a process of geometric Brownian motion without jumps and stochastic volatilities for three reasons. First, the standard model for the geometric Asian options has an analytical solution that is a reliable benchmark to verify the accuracy and speed of our method. Second, our method is essentially a model-free and data-driven method that can potentially learn option prices generated by models with a broad range of processes. Third, the stochastic processes characterizing the underlying stock prices do not affect our method since our method is independent of the option pricing models. The model in our study serves the purpose of data generation for verifying the effectiveness of our model-free method.

Our work also connects to one of the developments of financial technology in applying machine learning technologies to financial prediction and asset pricing. Heaton et al. (2016) apply some deep learning algorithms of prediction and classification to discover the function relationship between a dependent variable and a group of independent variables that cannot be revealed by existing financial economic theory. McGhee (2018) applies neural networks to a general stochastic volatility process and achieves a high degree of accuracy and 10,000 times faster than the finite difference method. Liu et al. (2019) introduce an efficient method of neural networks for calibrating the parameters of high-dimensional stochastic volatility models by avoiding the issues caused by local minima. Horvath et al. (2019) employ a neural network to calibrate several volatility models and show that it only takes a few milliseconds to calibrate the full implied volatility surface. Weigand (2019) provides a literature review on the application of machine learning to empirical asset pricing with a highlight of the pitfalls in the application. More generally, Fan et al. (2019) present a survey on common neural network models and point out the practical and theoretical benefits of deep learning. Our work complements this strand of literature by comprehensively investigating the effectiveness of pricing Asian options with deep learning.

Our paper is most closely related to Culkin and Das (2017), who train a deep learning neural network to calculate standard European option prices, which can be directly obtained from the Black and Scholes formula. Our study is different from theirs in several aspects. First, we use deep learning to estimate the prices of both geometric and arithmetic Asian options, where the latter does not have an explicit formula. We successfully provide a new method with deep learning to solve the challenging problem. Second, we use the Adam optimization algorithm to update the model parameters in our neural network, which reaches more accurate results than other updating methods. Third, we perform a series of comprehensive random experiments to investigate the effects of the sample size on the accuracy. For each sample, we randomly carry out 100 times of deep learning using 10 different training sets and testing sets and 10 different initial states. These random experiments support the robustness of our results. Using these random experiments, we compare our method with the simulation method, the explicit formula, and the real data, which all verify its effectiveness. Last, we also examine computational efficiency and we reveal that the new method is not only more accurate but also much faster than the traditional methods including the analytic formula method for the geometric Asian options.

Our paper and Fang and George (2017) share some common interests in the application of machine learning to the pricing of Asian options, but our work differs from theirs in the aspects of methods and effectiveness. First, they integrate the classic Levy (1992) approximation formula for arithmetic Asian options with a single-layer neural network that acts as a filter to map real volatilities from data to implied volatilities for the Levy approximation. Their method is not a model-free method and it relies on the assumptions of Levy (1992) model. By contrast, our method is independent of any option pricing models and it directly applies a multi-layer deep learning neural network to discover a way of estimating option prices. Second, using WTI option data, the accuracy achieved by their method is not as ideal as their simulation experiments while our method achieves high accuracy in WTI data. Specifically, the order of magnitude of their MSE ranges from 10^{-3} to 10^{-1} while ours is from 10^{-6} to 10^{-7}. Their R^2-value is about 0.72 to 0.9942 while ours is about 0.99987. Due to model dependence, the performance of their method deteriorates in real data where there are large differences between real volatilities and implied volatilities. On the contrary, our method is not limited by any model assumptions and therefore it can learn real data effectively through deep learning.

The structure of the paper is as follows. Section 2 introduces the pricing problem of Asian options and three methods of option pricing are discussed: the analytical solution, the simulation method, and the deep learning method. Section 3 provides numerical experiments and empirical analysis to verify the new method based on deep learning. Section 4 summarizes the main findings.

2. The model

The pricing of financial derivatives and the construction of hedging strategies play an important role in financial economics. Among all derivatives, Asian options are popular and their claims depend on the average prices of underlying assets for a given period. It is difficult for speculators to change the payoff of Asian options by manipulating its underlying asset price near the maturity date and thus Asian options avoid some shortcomings of European options. There are two types of Asian options: arithmetic Asian options and geometric ones. This section firstly introduces the two types of options and presents their pricing methods. After that, a deep machine learning method is provided to price the Asian options.

2.1. Basic model settings

There are two types of assets listed in the financial market. One is the risk-free asset called a bond, whose price $B(t)$ at time t satisfies the following ordinary differential equation:

$$B(t) = rB(t)dt, \quad B(0) = 1, \quad 0 \leq t \leq T,$$

where the constant r denotes the risk-free interest rate and the constant $T > 0$ is the maturity date. The other asset is a risky one called stock and its price $S(t)$ satisfies the stochastic differential equation below:

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t), \quad S(0) = S_0, \quad 0 \leq t \leq T,$$

where $W(t)$ is a one-dimensional standard Brownian motion that captures the randomness and risk in the market. The constant μ is the expected return rate of the stock and σ is its *volatility* that characterizes the standard deviation of the stock return.

2.2. The geometric Asian option

At the maturity date T, the payoff $V(T)$ of the geometric Asian option is determined by the geometric average of stock prices in the time interval $[0, T]$. The payoff is

$$V(T) = \left[\exp\left\{\frac{1}{T}\int_0^T \ln S(u)du\right\} - K\right]^+,$$

where $K > 0$ is the strike price of the option that is stated on the option contract. It is well known that the fair price $V(t)$ of the option is determined by the risk-neutral expectation of the terminal payoff $V(T)$ discounted by the risk-free interest rate r. That is

$$V(t) = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}(V(T)|\mathcal{F}_t), \quad V(0) = V_0, \quad 0 \leq t \leq T,$$

where $\mathbb{E}^{\mathbb{Q}}$ represents the expectation under the risk-neutral probability $\mathbb{Q}$.

According to Yan (2018) among others, the price $V(t)$ of the geometric Asian option can be solved and represented by the following explicit expressions:

$$V(t) = e^{-r(T-t)}[\exp(\Gamma)\Phi(M) - K\Phi(N)], \quad 0 \le t \le T,$$

where $\Phi(\cdot)$ represents the standard normal distribution function and

$$\begin{aligned}\Gamma \equiv& [\int_0^t \ln S(u)du + (T-t)\ln S(t)]/T + (T-t)^2(2r-\sigma^2)/(4T) \\ &+ (T-t)^3\sigma^2/(6T)^2,\end{aligned}$$

$$\Theta \equiv [4\int_0^t \ln S(u)du + 4(T-t)\ln S(t) - 4T\ln K + (T-t)^2(2r-\sigma^2)]/(4\sigma),$$

$$M \equiv \frac{\Theta + \sigma(T-t)^3/(3T)}{(T-t)^{\frac{3}{2}}/\sqrt{3}}, \quad N \equiv \frac{\Theta}{(T-t)^{\frac{3}{2}}/\sqrt{3}}.$$

In particular, the price of the geometric Asian option at the initial time $t = 0$ is

$$V_0 = S_0\Phi(z + \sigma\sqrt{T}/\sqrt{3})\exp\{-(6rT + \sigma^2 T)/12\} - K\Phi(z)\exp\{-rT\},$$

where

$$z = [-4\sqrt{3}\ln K + 4\sqrt{3}\ln S_0 + \sqrt{3}T(2r - \sigma^2 T)]/(4\sqrt{T}\sigma).$$

The above expression shows that the value process of the geometric Asian option is independent of the expected return rate μ of the underlying asset (stock).

2.3. The arithmetic Asian option

The payoff $V(T)$ of an arithmetic Asian option at the terminal time T is

$$V(T) = \left[\frac{1}{T}\int_0^T S(t)dt - K\right]^+,$$

where $K > 0$ is the strike price of the option. It is very difficult to price the arithmetic Asian option, in sharp contrast to the geometric Asian option. It is time-consuming to approximate the option price by the traditional methods of numerically solving partial differential equations or Monte Carlo numerical simulation. Yang et al. (2011) provide explicit expressions for pricing the arithmetic Asian option but it still requires a series of time-consuming computations to obtain a specific price. These prior methods have to perform expensive and repetitive computations when some parameters are changed, which are impractical for real-time investment.

To approximate the arithmetic Asian option price by the traditional Monte Carlo simulation method, one needs to simulate a large number of paths of the underlying stock asset prices and option prices by a high-performance computer. Specifically, the simulation method starts from simulating a number of paths of the stock price and then it computes the arithmetic average of the asset prices on each path. After that, one computes the sample payoff of the option for each path. Finally, after discounting all sample payoffs to the initial time by the risk-free interest rate r and calculating their mean, we obtain an approximation of the fair price of the option.

2.4. Deep learning framework

For pricing Asian options, we apply a deep learning algorithm of Back Propagation (BP) neural network that comprises the forward propagation of a working signal and the back propagation of an error signal. The BP algorithm is implemented on the *TensorFlow* framework (Abadi et al., 2016), which is developed by Google for deep learning. The TensorFlow framework has the advantages of flexibility, efficiency, scalability, and portability.

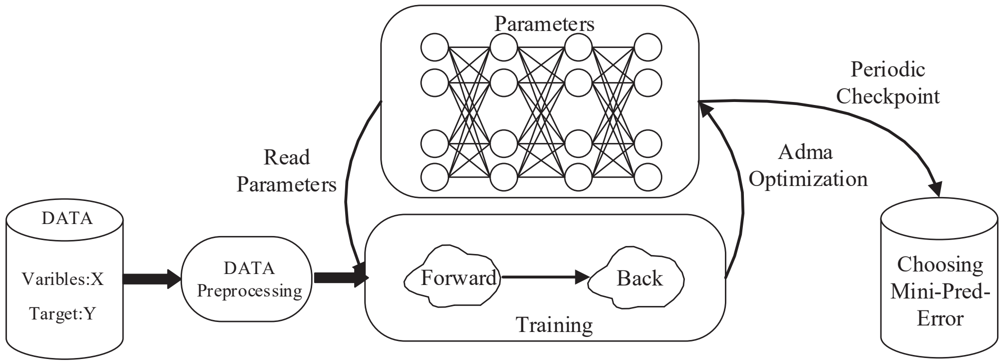


Figure 1. Flowchart of BP neural network deep learning process.

Figure 1 draws the flowchart of the BP neural network deep learning process. It comprises the process of data transmission and the process of parameter updating. The update process applies the Adam stochastic gradient descent optimization algorithm. The flowchart extends the general flowchart in Abadi et al. (2016) by specifying variables, deep learning, and the optimization method.

Deep learning refers to machine learning models that comprise multiple processing *layers* to analyze data. The first layer is the input layer and the last layer is the output layer. The layers between the input layer and the output layer are *hidden layers* that affect the complexity and effectiveness of a deep learning algorithm. In our implement of deep learning, we choose four hidden layers and each of them contains 100 *neurons.* These neurons receive inputs from previous neurons and then they process data with *activation* functions. The activation functions that we use on our four hidden layers are as Culkin and Das (2017): Leaky ReLU (Leaky Rectified Linear Unit, Maas and Ng (2013)), ELU (Exponential Linear Unit, Clevert et al. (2016)), ReLU (Rectified Linear Unit, Nair and Hinton (2010)), and ELU again, respectively.

Figure 1 portrays the deep learning model of the BP neural network, which comprises the process of data transmission and the process of parameter update. During the process of parameter updating, the model compares the predicted values and the output value of the network in order to compute the predicting error between the predicted values and the output value, which determines the value of a previously given *loss function*. Meanwhile, the model calculates the gradient of the loss function and propagates relevant parameters according to a chain rule. Our model uses the Adam (Adaptive moment estimation, Kingma and Ba (2014)) stochastic gradient descent optimization algorithm to update the model parameters.

Before the learning process starts, we need to set some hyper-parameters for the BP neural network. Following a typical setting in practice and the literature, e.g., Culkin and Das (2017), we set a dropout rate of 25% to avoid over-fitting data and the batch size of data for each training is 64. With these hyper-parameters, we let the model run training 2,500 times. For the Adam algorithm, we set the learning rate to 0.1 and keep the default values of the TensorFlow framework for other Adam parameters. Finally, we let the deep learning model output the results with the minimal *mean square error (MSE)* of prediction.

3. The effectiveness of pricing Asian options with deep learning

In this section, we demonstrate the effectiveness of the new option pricing method with deep learning through two different kinds of data. The first is the artificial data produced by the computer to train and to test the deep learning model. We let the computer generate three sets of option price data by three traditional methods: the explicit formula method of geometric

Asian options and the simulation method for both geometric Asian options and arithmetic Asian options. To obtain three sets of price data with a large size, we vary the parameter values of these Asian options within some ranges. After that, we divide the artificial data into two parts and we use one part of these data to train our BP neural network model and the other to test the prediction accuracy of the model. To check our method, we collect real Asian option data from a market. We use real data to further verify the effectiveness of the deep learning method for option pricing.

3.1. The effectiveness of deep learning by computer-generated data

In this section, we explain the effectiveness of deep learning by a simulation computation and it is further verified by an empirical analysis in the next section.

3.1.1. Data generation process

To generate a large size of data set, we first need to randomly draw option parameters from some chosen ranges of values. Table 1 lists the ranges of parameter values that we choose, which are similar to Culkin and Das (2017), who consider European options. In addition, we follow the convention of 250 trading days in a year. The annualized maturity is within the range of [0.004, 3] with a time interval being 1/250. It is also assumed that the strike prices of the Asian options are between 0.7 and 1.3 times the initial price of the underlying asset, which are taken in practice by most traders in the market.

After obtaining a random draw of the option parameter values from Table 1, we use three traditional pricing methods to generate three sets of artificial Asian option data, with which we examine the effectiveness of the

Table 1. Parameters and value ranges

Parameter	Range
Stock price (S)	\$10 - \$500
Strike price (K)	\$7 - \$650
Maturity (T)	1 day to 3 years
Risk free rate (r)	1% - 3%
Volatility (σ)	5% - 90%
Call price (C)	\$0 - \$328

Notes. Table 1 lists the ranges of option parameters that we use to generate option data for numerical comparison and analysis. We take a large number of random draws within these ranges in order to generate a large size of sample data.

new method based on deep learning. As shown in the flowchart Figure 2, the three traditional methods are the exact formula method for geometric Asian options, the geometric average method for geometric Asian options, and the arithmetic average method for arithmetic Asian options. For the last two methods, we use Monte Carlo simulation to produce the stock price path 5,000 times and we calculate the option payoff for each stock path. In total, we stimulate 5,000 payoffs. After discounting the payoffs to the initial time and calculating their mean, we generate one option price. In this way, we repeat the random draw, exactly analytic computation, and stochastic simulation a number of times to obtain three large sets of sample data.

Before feeding a set of option price data into a deep learning model, we standardize these data. The option pricing theory implies that the option price V is linearly correlated with the stock price S and the option strike

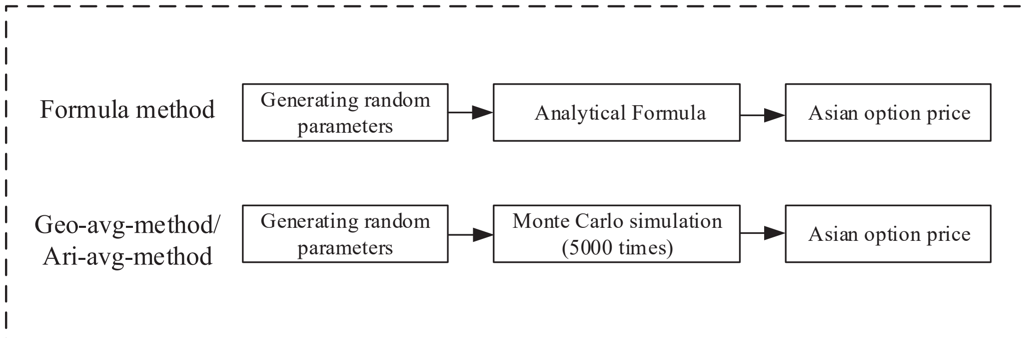


Figure 2. Flowchart of data generation processes.

Figure 2 draws the flowchart of data generation processes. We use three traditional methods: (1) the exact formula method for geometric Asian options, (2) the geometric average method for geometric Asian options, and (3) the arithmetic average method for arithmetic Asian options.

price K, see Hutchinson et al. (1994). Hence, one can standardize the data by dividing the option prices and the stock prices by the strike price K as follows:

$$V(S_0, K)/K = V(S_0/K, 1).$$

After that, one can input the standardized data along with the five parameter values S_0, K, T, r, σ into a deep learning model.

For the three sets of computer-generated data, we divided each set of data into a series of sample groups with six different sizes: 500, 1,000, 5,000, 10,000, 20,000 and 50,000. For each group of data, we randomly allocate data to a training set and a test set according to the ratio of 4:1. For example, the group of data with 500 option prices is randomly allocated into 400 training prices and 100 test prices. Then, we use these sample data to examine the effectiveness of the pricing method with deep learning. To prevent the contingency of the experimental results, we randomly generate the sample groups and allocate the sample group randomly into a training set and a test set. The process is repeated 10 times. For each group of data with an allocation of the training set and the test set, we use the data to train the deep learning model 10 times with 10 random initial states. In total, each group of sample is used to train the deep learning model 100 times under different allocations and initial states.

3.1.2. The accuracy analysis of numerical results

As described before, we have three sets of computer-generated data of Asian option prices using three traditional methods: the explicit and exact formula for geometric Asian options, the simulation method for geometric Asian options, and the simulation method for arithmetic Asian options. For each set of data, we divide them into six groups with the sizes of 500, 1,000, 5,000, 10,000, 20,000 and 50,000. For each group of data, we carry out 100 times of deep learning using 10 different training sets and testing sets and 10 different initial states. Each deep learning provides the outputs of 31,001 parameter values, of which 30,600 are weights and 401 are bias parameters.

After 100 runs of deep learning, we report five measures for the effectiveness of deep learning: the bias median, the 95% bias mean, i.e. the mean of the predicted errors which are less than the 95th percentile, the mean square error (MSE), the correlation coefficient ρ between the original data and the predicted data, and the R^2 value for the training set and the test set of each data group, as shown in Table 2 to Table 4. The *bias* represents the *relative* prediction error. That is, the program firstly computes the absolute value of the difference between a predicted option price and the option price from the original data and then it divides the absolute difference by the output price.

Table 2. Effectiveness of deep learning using data generated by the exact formula of geometric Asian option price.

Sample Size	500	1,000	5,000	10,000	20,000	50,000
Training Set						
Bias Median	0.01100	0.01219	0.01245	0.01283	0.01268	0.01290
95% Bias Mean	0.02335	0.02667	0.02603	0.02663	0.02623	0.02665
MSE	6.17E-06	7.49E-06	8.7E-06	8.92E-06	8.72E-06	9.02E-06
ρ	0.99983	0.99870	0.99892	0.99836	0.99887	0.99955
R^2	0.99960	0.99952	0.99943	0.99942	0.99943	0.99941
Testing Set						
Bias Median	0.01348	0.01361	0.01265	0.01283	0.01283	0.01284
95% Bias Mean	0.02951	0.03205	0.02753	0.02683	0.02684	0.02668
MSE	1.19E-05	1.05E-05	8.96E-06	9.01E-06	8.88E-06	8.99E-06
ρ	0.99980	0.99793	0.99878	0.99840	0.99878	0.99947
R^2	0.99922	0.99927	0.99942	0.99942	0.99942	0.99941

Notes. Table 2 reports five measures for the effectiveness of pricing geometric Asian options by the deep learning method under six groups of sample data that are generated by the exact formula of geometric Asian option price.

Table 2 reports five measures for the effectiveness of pricing geometric Asian options by the deep learning method under six groups of sample data that are generated by the exact formula of geometric Asian option prices. We find that these measures are robust across the six groups of data for both the training set and the testing set. Take the training set for example. The

Table 3. Effectiveness of deep learning using data generated by the simulation of geometric Asian option price.

Sample Size	500	1,000	5,000	10,000	20,000	50,000
			Training Set			
Bias Median	0.02536	0.02546	0.02587	0.02581	0.02623	0.02602
95% Bias Mean	0.05383	0.05478	0.05124	0.05113	0.05163	0.05298
MSE	7.62E-05	4.21E-05	3.88E-05	4.56E-05	4.13E-05	3.82E-05
ρ	0.99406	0.99672	0.99723	0.99596	0.99696	0.99811
R^2	0.99514	0.99727	0.99748	0.99705	0.99730	0.99751
			Testing Set			
Bias Median	0.02910	0.02765	0.02645	0.02581	0.02635	0.02612
95% Bias Mean	0.09776	0.05866	0.05361	0.05222	0.05173	0.05372
MSE	3.75E-05	3.23E-05	3.05E-05	3.67E-05	5.15E-05	4.07E-05
ρ	0.99460	0.99656	0.99750	0.99663	0.99653	0.99803
R^2	0.99761	0.99797	0.99801	0.99765	0.99662	0.99734

Notes. Table 3 reports five measures for the effectiveness of pricing geometric Asian options by the deep learning method under six groups of sample data that are generated by the simulation of geometric Asian option price.

medians of bias indicate that the relative errors of the predicted option prices in more than half of the training sets are within 1.4%. The 95% bias means are about 2.66%. The MSEs in the training set are near zero at 10^{-6}. The correlations ρ between the predicted data and the original data are almost 1, so are the values of R^2. In the testing set, all of these measures are similar to those in the training set, which explains that there are no over-fitting

Table 4. Effectiveness of deep learning using data generated by the simulation of arithmetic Asian option price.

Sample Size	500	1,000	5,000	10,000	20,000	50,000
			Training Set			
Bias Median	0.02418	0.02546	0.02617	0.02594	0.02639	0.02578
95% Bias Mean	0.04557	0.04776	0.04945	0.04797	0.04971	0.04799
MSE	3.51E-05	4.26E-05	4.63E-05	4.81E-05	5.67E-05	5.06E-05
ρ	0.99758	0.99773	0.99636	0.99584	0.99570	0.99540
R^2	0.99828	0.99788	0.99766	0.99754	0.99713	0.99740
			Testing Set			
Bias Median	0.02798	0.02724	0.02669	0.02610	0.02642	0.02584
95% Bias Mean	0.05648	0.05052	0.05146	0.04858	0.05144	0.04789
MSE	5.26E-05	4.70E-05	6.80E-05	5.87E-05	4.84E-05	5.19E-05
ρ	0.99402	0.99732	0.99559	0.99518	0.99561	0.99570
R^2	0.99742	0.99752	0.99645	0.99702	0.99759	0.99732

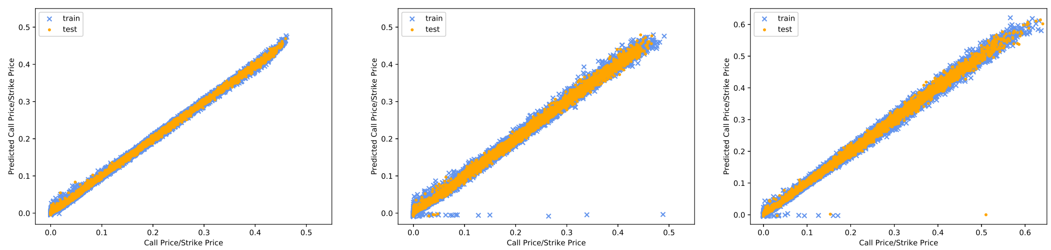
Notes. Table 4 reports five measures for the effectiveness of pricing arithmetic Asian options by the deep learning method under six groups of sample data that are generated by the simulation of arithmetic Asian option price.

problems in our model.

Similarly, Table 3 and Table 4 show the measures for the effectiveness of pricing geometric and arithmetic Asian options by the deep learning method under six groups of sample data. In contrast to Table 2, the two sets of data for Table 3 and Table 4 are generated by the simulation of geometric and arithmetic Asian option prices respectively. Over the two sets of data, the effectiveness of deep learning is robust across the six groups of data for both the training set and the testing set. The medians of bias indicate that the relative errors of the predicted option prices in more than half of the data are not above 3%. Almost all of the 95% bias means are less than 5.5%. The MSEs are near zero at 10^{-5}. The values of R^2 and the correlation ρs between the original data and the predicted data are almost 1.

Compared with Table 2, the medians of bias and the 95% bias means in Table 3 and Table 4 are about twice the corresponding values in Table 2. These differences are expected since the data to be learned for Table 2 are obtained from the analytical solution while the data that are inputted to the deep learning method for Table 3 and Table 4 are generated by simulation. Random numbers for simulation unavoidably introduce more noises to the simulation-generated data than the data obtained from the analytical solution. Although the performance implied by Table 3 and Table 4 is not as high as that indicated by Table 2, it is reasonable and acceptable. Indeed, the MSEs are kept at the level near zero and both R^2-value and ρ-value are close to 1 in Table 3 and Table 4 as well. In addition, the differences in these tables show that our method is a data-driven method and the data quality affects its performance. When we use actual option data to train the deep learning model in Section 3.2, we achieve lower biases than those based on the data generated by the analytical formula.

In short, from the results in Table 2 to Table 4 we conclude that our method is robust in both training sets and testing sets across different sizes of sample data generated by three kinds of methods. These robust results demonstrate the effectiveness of the deep learning method for pricing Asian

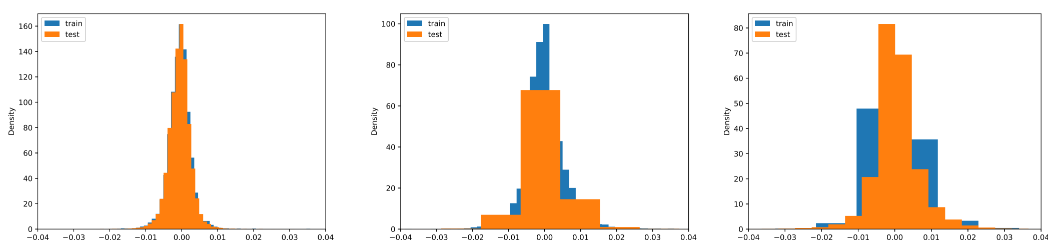


(a) (b) (c)

Figure 3. Predicted prices vs computer generated prices of Asian options.

Figure 3 plots the predicted prices using deep learning vs the computer generated prices using (a) the exact formula of geometric Asian option; (b) the simulation of geometric Asian option; (c) the simulation of arithmetic Asian option.

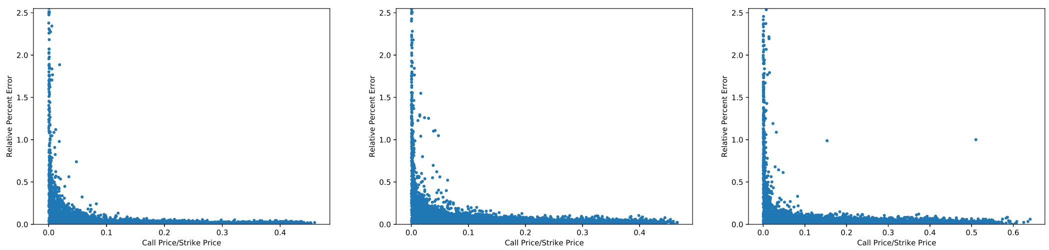
options even if the size of the training data is small. We emphasize that the robust results across different sizes of sample data are particularly useful in practice since the size of real data for one particular option is usually limited.



(a) (b) (c)

Figure 4. Absolute prediction errors of Asian option prices.

Figure 4 plots the densities of absolute prediction errors between the predicted prices using deep learning and the computer generated prices using (a) the exact formula of geometric Asian option; (b) the simulation of geometric Asian option; (c) the simulation of arithmetic Asian option.



(a) (b) (c)

Figure 5. Relative prediction errors of Asian option prices.

Figure 5 plots the relative prediction errors between the predicted prices using deep learning and the computer generated prices using (a) the exact formula of geometric Asian option; (b) the simulation of geometric Asian option; (c) the simulation of arithmetic Asian option.

After we show the robust results of deep learning across different sizes of data, we illustrate the effectiveness of deep learning intuitively in Figure 3 to Figure 5. Similar to Culkin and Das (2017) who examine standard options, we plot the prediction prices, which are standardized by the corresponding strike prices, and prediction errors for Asian options under three different situations. In each of these figures, we use three traditional methods to generate three sets of data. The sub-figures with the label "(a)" use the data sets generated by the exact formula for geometric Asian options, while the sub-figures "(b)" and "(c)" take the data sets from the simulation of geometric and arithmetic Asian options respectively. We use the groups of data with the size of 50,000 across the three sets of data to produce the figures.

Figure 3 displays the predicted prices using deep learning vs the price data generated by the computer according to three traditional methods. It shows that for both training set and testing set, almost all of the price pairs are close to a straight line at the 45 degrees with a very narrow width, which means that the predicted prices are close to the option price data that we input into the deep learning model.

Figure 4 plots the distributions of the absolute prediction errors to investigate the errors between the standardized predicted option prices using deep learning and the price data generated by the three traditional methods. We find that most of the pricing errors for both training set and testing set under three sets of data are within ± 0.02.

To highlight how the predicted relative errors change with the ratios of the option prices to the strike price, Figure 5 depicts the standardized relative pricing errors of the deep learning method using the testing sets of data generated by the three methods mentioned above. It states that for most of the cases in the three sub-figures, the relative pricing errors of deep learning are quite low, except for the case where the ratios of the option prices to the strike prices near zero. If the ratio is close to zero, we get a large relative error.

3.1.3. The efficiency analysis of numerical experiments

The above analysis discusses the accuracy of the deep learning method of pricing Asian options. Finally, we examine the efficiency of the deep learning method. Table 5 compares the time of computing 1,000 or 10,000 prices of Asian options by four methods: the deep learning method represented by D.L., the exact formula method of geometric Asian options, the simulation method of geometric Asian options, and the simulation method of arithmetic Asian options.

Table 5. The comparisons of computation time

No. of Prices	**Geometric**		**Geometric**		**Arithmetic**	
	D.L.	**Formula**	**D.L.**	**Sim.**	**D.L.**	**Sim.**
Panel A: Computation Time Using a Laptop						
1,000	0.54s	2.43s	0.50s	$> 10^4$s	0.48s	$> 10^4$s
10,000	0.76s	22.16s	0.72s	$> 10^5$s	0.60s	$> 10^5$s
Panel B: Computation Time Using a Workstation						
1,000	0.13s	0.27s	0.13s	$\approx 10^4$s	0.15s	$\approx 10^4$ s
10,000	0.22s	2.70s	0.22s	$\approx 10^5$s	0.22s	$\approx 10^5$ s

Notes. Table 5 compares the time of computing 1,000 or 10,000 prices of Asian options by four methods: the deep learning method represented by D.L., the exact formula method of geometric Asian options, the simulation method of geometric Asian options, and the simulation method of arithmetic Asian options. Panel A and B list computational time using a laptop and a workstation respectively. "$> 10^n$" ("$\approx 10^n$") represents that the computation time is greater than (within) the order of magnitude 10^n.

Noting that the deep learning method can be applied by individual investors or institutional investors, we report computational time using an ordinary laptop in Panel A and a high-performance workstation in Panel B. Besides, all of the other results in this paper are obtained by employing the workstation. The laptop hardware specifications include a CPU of Intel® Core™ i5-5200U Processor @ 2.20 GHz, a GPU of NIVIDA GeForce 840M, a RAM of 4GB, and an HDD of 500GB. The workstation hardware specifications are two CPUs of Intel® Xeon® E5-2699 v4 @ 2.20 GHz, a GPU of NIVIDA Quadro M6000, a RAM of 256G, and four SSDs with 10TB in total.

Table 5 states that the time spent by the trained deep learning model in computing 10,000 geometric or arithmetic Asian option prices is almost the same and is less than 1 second even with an ordinary laptop. The deep learning method is even much faster than the exact formula method, which takes 2 to 22 seconds by using a common laptop, let alone the time-consuming simulation method, which spends 10,000 seconds to 100,000 seconds. We emphasize that the computation speed is key for practitioners in the industry to succeed in trading since they usually need to carry out a large number of computations or tests in a short period and make prompt decisions in a fast-changing financial market.

3.2. The effectiveness of deep learning by real data

To verify the effectiveness of deep learning further, we consider the real transaction data of Asian options downloaded from a financial market. The data about Asian options are limited and it is difficult to obtain a large number of relevant data since Asian options are usually non-standardized over-the-counter financial products.

We use the Light Sweet Crude Oil (WTI) Futures and Options data at Barchart.com. WTI stands for West Texas Intermediate, which is a light sweet crude oil stream that comprises a mix of several streams of light sweet crude oil in the U.S. We use the deep learning option pricing method to learn the data of WTI Average Price Options. The underlying asset is the WTI futures.

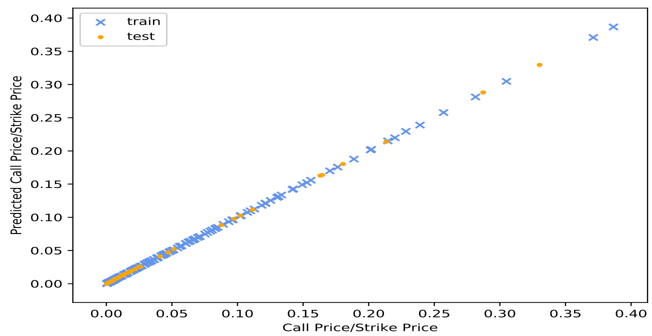
We download the data in early June of 2019 and choose the average options labeled by CLN19 (expired after one month), CLU19 (expired after three months), CLF19 (expired after half a year) and CLM20 (expired after one year). We select the option strike prices that are 0.7 to 1.3 times of the underlying asset prices. In addition, we take the corresponding implied volatilities near these maturities and the LIBOR interest rates corresponding to these maturities. We are only able to acquire 162 actual option prices but learning is still powerful as shown below.

Table 6. Effectiveness of deep learning by real data

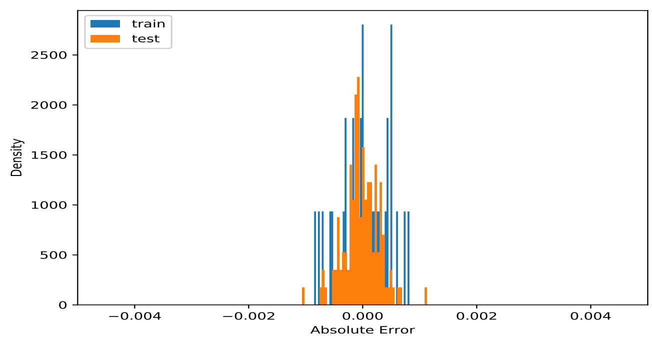
Data Source	Ari. Sim.	Geo. Sim.	Geo. Formula	Real Data
Training Set				
Bias Median	0.02207	0.01932	0.00930	0.00509
95% Bias Mean	0.04905	0.03412	0.01932	0.01758
MSE	2.01E-05	1.39E-05	4.67E-06	1.56E-07
ρ	1.00078	0.99901	0.99976	1.00010
R^2	0.99896	0.99919	0.99970	0.99997
Testing Set				
Bias Median	0.03476	0.02625	0.01308	0.00811
95% Bias Mean	0.04150	0.03723	0.02557	0.02015
MSE	6.40E-05	4.60E-05	1.49E-05	1.97E-06
ρ	1.02258	0.98960	1.00730	0.99526
R^2	0.99630	0.99756	0.99924	0.99987

Notes. Table 6 compares the effectiveness of pricing Asian options by the deep learning method using four different sources of data: arithmetic Asian options with simulation, geometric Asian options with simulation, geometric Asian options with the exact formula, and real data of market prices.

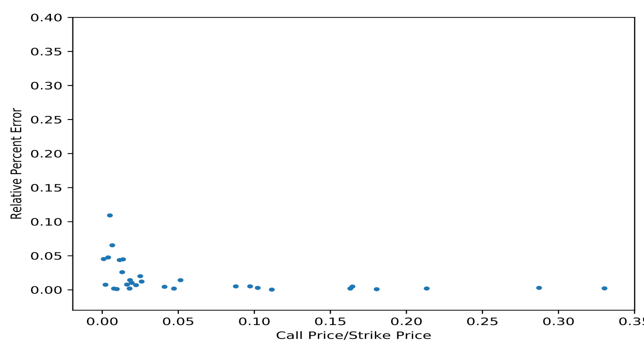
In the previous text, we demonstrate that pricing Asian options by the



(a)



(b)



(c)

Figure 6. Prediction of Asian option prices using real data.

Figure 6 plots (a) the predicted prices using deep learning vs the real prices; (b) the absolute prediction errors; (c) the relative prediction errors, between the predicted prices and the real prices of Asian options.

deep learning method is powerful across the sample sizes of 500, 1,000, 5,000, 10,000, 20,000 and 50,000 respectively. For comparison, we naturally wonder what the effectiveness of the experiments is if the sample size is 162, i.e. the size of the sample we obtain from the real market. For this reason, we repeat here the previous computations but take the sample size as 162 while we conduct the empirical analysis. With this sample size, we keep the ratio of the training set to the test set as 4:1. That is, we allocate the sample data into the training set with a size of 130 and the testing set with a size of 32.

Table 6 shows that all of the five measures for the effectiveness of deep learning by real data are superior to those of deep learning by the computer-generated data. The superior performance is robust in both the training set and the testing set. Specifically, the medians of bias indicate that more than half of relative errors of the predicted option prices in the real data are less than 0.8%. The 95% bias means, i.e. the means of the predicted errors which are less than the 95th percentile, are less than 2%. The MSEs are near zero at 10^{-6}. The values of R^2 and the correlations ρ between the real data and the predicted data are almost 1.

A 2% bias in our empirical results is acceptable in industry practice and reasonable in the empirical study of option pricing.[2] First, in industry practice, a 2% bias in the price of an option would not be a substantial issue to practitioners who take large positions in the option. Take hedge funds for example, which often hold large positions in options including Asian options. Using a data of 1,500 hedge funds, Gupta and Liang (2005) analyze

[2]We are grateful to an anonymous reviewer who points out the practical relevance of a 2% bias.

the Value-at-Risk (VaR) relative to fund assets, which measures the relative amount of capital that is required to cover most of the potential losses given a confidence level, and the capital adequacy (*Cap*), which is the ratio of the actual extra capital over the required capital. They report that the median (mean) relative VaR of the live funds is 9.8% (11.3%) and their median (mean) *Cap* ratio is 2.4 (5.3). The two measures imply that if a 2% bias in option pricing would lead to losses, the losses are acceptably covered by option holders with large positions. Therefore, a 2% bias would not cause a substantial issue in industry practice.

Second, in the empirical study of option pricing, a 2% bias is reasonably small. The option pricing error between the theoretical/estimated price and the market price has long been examined in the literature. Merton (1976) investigates the effects of model specification on option pricing and stock returns and he specifies a criterion of 5%, which is more than double of our 2% bias. Based on S&P 500-stock index options, Fortune (1996) discovers systematic and sizable errors that are produced by the widely-used Black-Scholes option pricing model. There are average 10% to 100% (15% to 40%) pricing bias for call (put) options. Similarly, Yakoob and Durham (2002) show pricing bias of 0.060% to 70.684% (0.416% to 29.118%) for call (put) options. More recently, Heo et al. (2017) find pricing biases ranging from about 2.28% to 4% using six models of European / American options and Yahoo options data. The potential explanations for these biases are limitations on arbitrage, short-selling restrictions, and unrealistic model assumptions. Compared with these existing studies, the 2% bias from the deep learning pricing method in our study is reasonable.

To illustrate the effectiveness intuitively, Figure 6 compares the standardized prediction prices of deep learning and the standardized real prices of average options. Figure 6(a) shows that the predicted prices and the real prices are close to a line approaching 45 degrees with a very narrow width, which indicates that the prediction of deep learning in actual data is excellent and the errors in both training set and test set are small. Indeed, Figure 6(b) shows that most of the absolute pricing errors compared with real data are between ± 0.0015 in both training set and testing set. Similarly, Figure 6(c) displays that the relative pricing errors of standardized prices are quite small, except for a small number of cases where the option values near zero, naturally leading to large relative errors.

The results in Figure 6 shows that the performance of deep learning by real data is better than the performance of deep learning by the data generated by the computer. There are two reasons for the advantage of deep learning by real data. First, in the real market, there are barely transactions near extreme situations where the standardized option price C/K approximates 0. As pointed out before, the standardized option values near 0 push up the relative pricing errors. Second, the option parameter values in the real data must be distributed in some short intervals instead of the large ranges in the numerical experiment. Therefore, Figure 6 illustrates that the learning process using real data is much more effective than using the data generated by the computer.

Last, we emphasize that to obtain the superior results in Figure 6, the empirical test conducted here does not require any assumptions on the probability distribution of the underlying asset, such as the common log-normal distribution. It only needs to correctly specify model parameters which impact on the option price and to collect sufficient sample data. Therefore, our machine learning method provides a new model-free approach for pricing financial assets. The model-free approach not only solves the challenges of pricing assets like arithmetic Asian options without closed-form solutions but also solves the pricing problems with closed-form solutions, e.g., pricing geometric Asian options, through an alternative way that avoids controversial model assumptions for the closed-form solutions. The implication from Figure 6 demonstrates that machine learning has a very bright future in financial applications.

4. Conclusion

In this paper, we use a deep learning model to predict Asian option prices and we examine the effectiveness and efficiency of the deep learning method by performing a numerical experiment and an empirical test with real data. In the numerical experiment, we investigate the effectiveness by using three sets of data that are generated by the computer according to three types of traditional methods: the exact formula of geometric Asian options, the simulation of geometric Asian options, and the simulation of arithmetic Asian options.

The numerical results and empirical analysis show that no matter which set of data is used to train the deep learning model, it can predict the Asian option prices with high accuracy. Compared with the three traditional methods, the speed of the trained deep learning model is extremely fast. To verify the feasibility of the deep learning method in practice, we use a set of real data about WTI Average Price Options to train the deep learning model, which produces more accurate results than those in the numerical experiment using our three sets of simulation data. The deep learning method achieves superior performance in the real data because there are fewer extreme cases in the real data than in the computer-generated data in our numerical experiments. Our numerical results imply that the deep learning method is expected to be robust to be applied in practice since our numerical experiments are tougher than the real situation. Furthermore, the deep learning method is a model-free approach for asset pricing, which avoids non-realistic model assumptions.

Artificial intelligence enters a new era after the on-going improvement of computer performance and the wide applications of machine learning. The applications of artificial intelligence in many fields have achieved remarkable results but their applications in finance just started. In finance, there are many tasks of finding a functioning relationship between a dependent variable and a series of independent variables from finance data in real-time. Generally, traditional methods are not capable of fulfilling such tasks. In this paper, we provide a method of applying the recent development in machine learning to price a class of financial products. We demonstrate that the method is effective as other applications of machine learning in other areas. This method can be conveniently applied by investment managers or traders in the industry of financial trading in the real world. With the development of computer techniques and big data, artificial intelligence in finance has a much brighter future than we expected.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al., 2016. Tensorflow: A system for large-scale machine learning. operating systems design and implementation, 265–283.

Cao, J., Chen, J., Hull, J. C., 2018. A neural network approach to understanding implied volatility movements. Available at SSRN 3288067.

Clevert, D., Unterthiner, T., Hochreiter, S., 2016. Fast and accurate deep network learning by Exponential Linear Units (ELUs). International Conference on Learning Representations (ICLR).

Culkin, R., Das, S. R., 2017. Machine learning in finance: the case of deep learning for option pricing. Journal of Investment Management 15 (4), 92–100.

Fan, J., Ma, C., Zhong, Y., 2019. A selective overview of deep learning. arXiv preprint arXiv:1904.05526.

Fang, Z., George, K. M., 2017. Application of machine learning: An analysis of Asian options pricing using neural network. In: The Fourteenth IEEE International Conference on e-Business Engineering. IEEE, pp. 142–149.

Ferguson, R., Green, A., 2018. Deeply learning derivatives. arXiv preprint arXiv:1809.02233.

Fortune, P., 1996. Anomalies in option pricing: The Black-Scholes model revisited. New England Economic Review, 17–41.

Fu, W., Hirsa, A., 2019. A fast method for pricing American options under the variance gamma model. arXiv preprint arXiv:1903.07519.

Fusai, G., Roncoroni, A., 2007. Implementing models in quantitative finance: methods and cases. Springer Science & Business Media.

Gupta, A., Liang, B., 2005. Do hedge funds have enough capital? A value-at-risk approach. Journal of Financial Economics 77 (1), 219–253.

Halperin, I., 2017. QLBS: Q-learner in the Black-Scholes(-Merton) worlds. arXiv preprint arXiv:1712.04609.

Heaton, J., Polson, N. G., Witte, J. H., 2016. Deep learning in finance. arXiv preprint arXiv:1602.06561.

Heo, S. W., Cashel-Cordo, P., Rhim, J. C., Kang, J. G., 2017. Pricing accuracy of put-option valuation models: Directional bias due to risk free interest rates. Journal of Accounting and Finance 17 (5).

Horvath, B., Muguruza, A., Tomas, M., 2019. Deep learning volatility. Available at SSRN 3322085.

Hutchinson, J. M., Lo, A. W., Poggio, T., 1994. A non parametric approach to pricing and hedging derivative securities via learning networks. Journal of Finance 49 (3), 851–889.

Jacquier, A., Malone, E. R., Oumgari, M., 2019. Stacked Monte Carlo for option pricing. arXiv preprint arXiv:1903.10795.

Karatas, T., Oskoui, A., Hirsa, A., 2019. Supervised deep neural networks (DNNs) for pricing/calibration of vanilla/exotic options under various different processes. arXiv preprint arXiv:1902.05810.

Kingma, D., Ba, J., 2014. Adam: A method for stochastic optimization. Computer Science.

Kolb, R. W., Overdahl, J. A., 2010. Financial derivatives: pricing and risk management. Vol. 5. John Wiley & Sons.

Levy, E., 1992. Pricing European average rate currency options. Journal of International Money and Finance 11 (5), 474–491.

Liu, S., Borovykh, A., Grzelak, L. A., Oosterlee, C. W., 2019. A neural network-based framework for financial model calibration. arXiv preprint arXiv:1904.10523.

Maas, Andrew L, H. A. Y., Ng, A. Y., 2013. Rectifier nonlinearities improve neural network acoustic models. Conference on Machine Learning (ICML) 30.

McGhee, W. A., 2018. An artificial neural network representation of the SABR stochastic volatility model. Available at SSRN 3288882.

Merton, R. C., 1976. The impact on option pricing of specification error in the underlying stock price returns. Journal of Finance 31 (2), 333–350.

Nair, V., Hinton, G. E., 2010. Rectified linear units improve restricted Boltzmann machines. Conference on Machine Learning (ICML), 807–814.

Vecer, J., 2001. A new PDE approach for pricing arithmetic average Asian options. Journal of computational finance 4 (4), 105–113.

Weigand, A., 2019. Machine learning in empirical asset pricing. Financial Markets and Portfolio Management, 1–12.

Yakoob, M. Y., Durham, N., 2002. An empirical analysis of option valuation techniques using stock index options. The working study, Duke University, Durham.

Yan, J.-A., 2018. Introduction to Stochastic Finance. Springer.

Yang, Z., Ewald, C.-O., Menkens, O., 2011. Pricing and hedging of Asian options: quasi-explicit solutions via Malliavin calculus. Mathematical Methods of Operations Research 74 (1), 93–120.