

Article

Sensitivity Analysis of Optimal Commodity Decision Making with Neural Networks: A Case for COVID-19

Nader Karimi ¹, Erfan Salavati ^{1,*}, Hirbod Assa ² and Hojatollah Adibi ¹ 

¹ Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran 1591634311, Iran

² Kent Business School, Canterbury CT2 7FS, UK

* Correspondence: erfan.salavati@aut.ac.ir

Abstract: The COVID-19 pandemic caused a significant disruption to food demand, leading to changes in household expenditure and consumption patterns. This paper presents a method for analyzing the impact of such demand shocks on a producer's decision to sell a commodity during economic turmoil. The method uses an artificial neural network (ANN) to approximate the optimal value function for a general stochastic differential equation and calculate the partial derivatives of the value function with respect to various parameters of both the diffusion process and the payoff function. This approach allows for sensitivity analysis of the optimal stopping problem and can be applied to a range of situations beyond just the COVID-19 crisis.

Keywords: optimal stopping; deep learning; sensitivity analysis; value function

MSC: 62P05; 97M30; 68Q32



Citation: Karimi, N.; Salavati, E.; Assa, H.; Adibi, H. Sensitivity Analysis of Optimal Commodity Decision Making with Neural Networks: A Case for COVID-19. *Mathematics* **2023**, *11*, 1202. <https://doi.org/10.3390/math11051202>

Academic Editors: Carmen Boado Penas, Julia Eisenberg and Sule Sahin

Received: 1 January 2023

Revised: 15 February 2023

Accepted: 20 February 2023

Published: 28 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The COVID-19 pandemic had a significant and uneven impact on household spending in the UK and the US, with lock-downs and other public health measures leading to a 22.2% decrease in UK household spending (This is based on information that can be extracted on the webpage of the UK's office for national statistics on COVID-19: <https://www.ons.gov.uk/> accessed on 30 December 2022) and a 60% shock in a short time in the US consumption of many services and goods (The data can be extracted from the US economic recovery tracker at <https://tracktherecovery.org/> accessed on 30 December 2022). However, some goods and services have experienced positive shocks. This has led us to investigate the impact of these changes in consumer demand on the optimal decision-making of producers. To the best of our knowledge, this is the first study to examine this issue. There have been numerous studies on changes in the consumption of various goods during the COVID-19 pandemic. While it might be expected that overall consumption would decrease, the actual patterns have been somewhat varied. For example, the consumption of fast food, bread, meat products, and beverages has generally declined, while the consumption of dairy and breakfast products, vegetables, fruits, and nutritional supplements has increased [1]. In the UK, there has been an increase in the consumption of high energy density snack and home-prepared foods, as well as fruits and vegetables [2]. Other studies have also examined changes in consumption and eating habits [3–6]. Despite all these studies, there is still a gap in our knowledge with regard to the changes in the behavior of the commodity market decision-makers. The main reason for this knowledge gap is that decisions are not observable and suitable models need to be employed to give us a better understanding of the situation.

To analyze these issues, the paper employs a storage model, which is a widely used formalism for modeling the optimal time to sell commodities. This approach is based on a large body of literature in the field. The storage models for commodity prices date back to [7], and were further developed by incorporating rational expectations in [8,9]. Refs. [10–12]

developed a partial equilibrium structural model of commodity price determination and applied numerical methods to test and estimate the model parameters. More recently, many authors improved the storage model in order to better capture the statistical characteristics; see for instance [13–19]. Particularly, in the last paper, the focus is on the analysis of storable commodities from the perspective of the optimal stopping problem. However a main challenge that we need to overcome in this paper is how the model behavior changes when there is a shock to the demand-related parameters in the model. This is exactly the situation we have observed during the COVID-19 economic lock-down. To the best of our knowledge, such a sensitivity analysis has not yet been studied in the literature.

In this paper, we can make different measurements of different important objects. That includes sensitivity and impulse analysis of the value function, in addition to the probability of the storage time by the the decision maker. The results are useful for the stress testing of the commodity market at the time of large economic events such as the COVID-19 economic downturn. For instance, we will see if the demand uncertainty increases, specifically the demand volatility, then the probability of storage is higher for larger commodity values.

Our approach to calibrating the model is to employ the deep neural network formalism. This approach has been used for optimal stopping time in American option pricing in [20,21]. However, the literature is mainly developed under the assumption that the model parameters are given. This is not what we can assume as we are concerned with the model sensitivity analysis with regards to the changes in the parameters of the model. Motivated by the literature, we use two demand models, namely exponential the OU and CIR models (see [18,19]). The implementation of such a sensitivity analysis needs new developments of the calibration method, which also included the parameters. Therefore, the parameters have to be fed to the machine as part of the learning process. This constitutes the major innovation that differentiates our paper from the technical perspective of other papers in the literature. We need to mention that using sensitivity analysis by neural networks is a popular approach in the literature, for example see [22].

The organization of this paper is as follows. In section 2, we provide a background of the technical account of the problem. In Section 3, the problem will be formulated in more detail. Section 4 introduces the neural network-based method for solving the optimal stopping problem. The application of this method to commodity markets is presented in Section 4. In addition, we present the sensitivity and impulse analysis of the optimal stopping problem using the proposed method. Finally, the paper concludes in Section 5.

2. Background

In this paper, we aim to understand the sensitivity and impulsiveness of the demand model to changes in various parameters. This is a challenging problem because any change in a parameter and impulsing demand can affect the optimal stopping time. To address this issue, we propose using deep neural networks as a means of providing a solution. Sensitivity and impulse analysis of the optimal stopping problem has not been widely studied in the literature. One example of such a study is [23], which uses analytical formulas for the value function to derive the signs of the derivatives of stopping boundaries with respect to various model parameters. However, this approach is limited to the simple Black–Scholes model and cannot be generalized to other models due to the lack of analytical solutions for the value function. Another study [24] examines the dependence of the value function on the initial condition and demonstrates the continuity of the value function, but this falls short of a comprehensive sensitivity and impulse analysis, which would require the differentiability of the value function. Ref. [25] uses neural networks to solve the optimal stopping problem and then calculates the Greeks of an American option using finite difference methods, rather than leveraging the computational capabilities of the neural networks for this task. Overall, the sensitivity and impulse analysis of the general optimal stopping problem remains a difficult problem.

There has been a recent trend of using deep learning in the optimal stopping problem. For example, ref. [20] proposes a deep neural network for directly learning the optimal

stopping time. One challenge in using neural networks for optimal stopping is that the stopping time is a discrete quantity, which makes it difficult to apply gradient-based methods, the main tool for training neural networks. To overcome this, ref. [20] breaks the stopping time into stopping decisions made at each time step, and then relaxes the 0-1 nature of these decisions by allowing them to take values within $(0, 1)$. This approach allows the stopping time to be represented as a function of the state of the process, which is approximated by the neural network. The authors also apply their method to pricing American-style options. Similar to many other works in the literature, ref. [20] uses simulations to generate inputs for training the neural network. Another approach to using neural networks in optimal stopping is taken by [21], which uses the Longstaff–Schwartz algorithm and approximates the conditional expectation $\mathbb{E}(G_{\tau_{n+1}}|X_n)$ for a payoff process G_t using a neural network. This approach is also extended to a hedging strategy, with the mean squared error of the conditional expectation serving as the loss function. In [26], a similar approach is taken with [20] with minor differences in the training phase. While [20] optimizes the stopping decision at each time step in a step-wise and recursive manner, ref. [26] optimizes the entire set of parameters for all time steps using a single objective function. Ref. [25] proposes a new neural network architecture that is trained only on the last layer, with the earlier layers initialized randomly. This architecture is used to approximate the continuation value of the optimal stopping problem and is claimed to be appropriate for high-dimensional data. The authors also use this method to calculate the Greeks of an American option, but rely on finite difference methods rather than neural networks for this purpose.

In this article, the focus is on the optimal stopping problem in a continuous-time framework. We propose using a neural network to approximate the optimal stopping probabilities and incorporate the model parameters as inputs to the network. This allows for efficient sensitivity and impulse analysis, as the complex functional form of the continuation value is avoided. The proposed method is applied to commodity prices, as the optimal stopping problem is particularly relevant in the commodities market where storage of the commodity is possible.

3. Formulation of the Problem

In this article, we are concerned with the optimal stopping problem for a stochastic process X_t as follows. First, we introduce the following stochastic differential equation:

$$dY_t = \mu(Y_t; \theta)dt + \sigma(Y_t; \theta)dB_t, \quad Y_0 = y_0, \quad (1)$$

where B_t is a Brownian motion and μ and σ are given functions parameterized by a set of parameters denoted by $\theta \in \mathbb{R}^d$. Then, we introduce $X_t = (1 + h)Y_t$. The additional impulse parameter h gives a shock to the system by replacing a process Y_t by X_t , where h is the percentage of the changes.

We assume by stopping at time τ that the agent will receive a payoff $e^{-r\tau}P(X_\tau; \theta)$ where r is the discounting factor and P is the instant payoff that depends on the parameters as well. Hence, the optimal stopping problem is to solve the following maximization:

$$\sup_{0 \leq \tau \leq T} \mathbb{E}[e^{-r\tau}P(X_\tau; \theta)],$$

where T is the terminal time and the supremum is taken over all stopping times τ in $[0, T]$.

Because of the Markovian nature of the process X_t , the optimal value of this stopping problem will depend only on the current time and the current state of the process rather than its history. In other words, we can define a value function:

$$V(t, x; \theta, h) = \sup_{t \leq \tau \leq T} \mathbb{E}[e^{-r(\tau-t)}P(X_\tau; \theta)|X_t = x]$$

Note that, by definition, $V(t, x; \theta, h) \geq P(x; \theta)$. The stopping region and continuation region are two important sets corresponding to an optimal stopping problem and are defined as follows:

$$D(t) = \{x | V(t, x; \theta, h) = P(x; \theta)\}, \quad C(t) = \{x | V(t, x; \theta, h) > P(x; \theta)\}.$$

Note that both C and D are implicitly dependent on θ and h . Now, the optimal stopping time τ^* can be described as follows:

$$\tau^* = \inf\{t \geq 0 : X_t \in D(t)\}$$

The aim of this article is to provide an efficient method to analyze the sensitivity of τ^* and V to the parameters θ .

We train an Artificial Neural Network (ANN) for computing the optimal stopping time. First, we discretize the time dimension in order to be able to feed the state process into the ANN. We consider $t_i = iT/N, i = 0, \dots, N$ be the time discretization. For simplicity, we will denote the value of process at time t_i , i.e., X_{t_i} , by X_i . The optimal stopping time in the new discrete setting takes one of the values t_i . To find τ , it suffices to determine at each time t_i whether to stop or to continue the process. This can be modeled as a random variable:

$$\xi_i = \begin{cases} 1, & \tau^* = t_i \\ 0, & \tau^* \neq t_i \end{cases}.$$

We will denote the probability of $\xi_i = 1$ by u_i , which will be the intended output of the ANN. Note that u_i is a function of X_i and θ . Hence, the ANN should be able to compute functions $u_i(x; \theta, h)$, $i = 0, \dots, N$. Given the value of u_i , the stopping rule can be determined as follows. We need to specify a confidence level γ (e.g., 0.95), which is a threshold for u_i , so that the stopping time is defined as below:

$$\tau^* = \min\{u_i; u_i \geq \gamma\}.$$

The ANN that we propose consists of $N + 1$ sub-networks indexed by $i = 0, \dots, N$ corresponding to each u_i , $i = 0, \dots, N$. The i th sub-network is intended to calculate u_i . The architecture of the sub-networks is as follows. Each sub-network has in itself three hidden layers, and each layer consists of 10 neurons. The input layer of each sub-network consists of $d + 1$ input neurons. One of the input neurons is reserved for the state of the process X_i and the other d neurons are used to feed the parameters, θ and h , into the i th sub-network. The activation functions for all hidden layers are chosen to be the ReLU function, but the activation function of the output layer is chosen to be the sigmoid function because the output represents a probability and should be in $(0, 1)$. We denote full set of weights and biases of ANN by a vector w . The output of the i th subnetwork is represented as $u_i(x; \theta, h; w)$. Figure 1 shows the full architecture of the ANN.

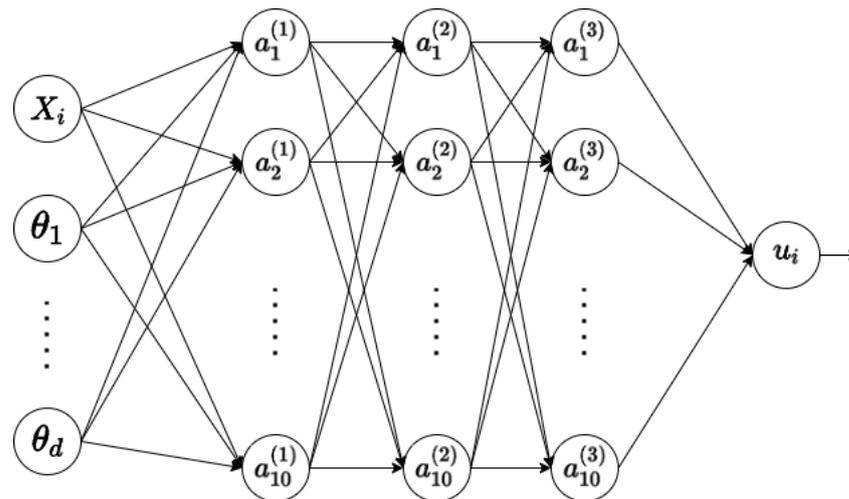


Figure 1. Three hidden layer neural network for computation of u_i .

Now, we explain the objective function of the ANN and the optimization method. A natural choice for the objective function is the expected payoff of the agent, which can be calculated as follows.

$$J(\theta, h; w) = \mathbb{E}[\tilde{J}(X; \theta, h; w)],$$

where \tilde{J} is the payoff of the agent given a sample path X , i.e.,

$$\tilde{J}(X; \theta, h; w) = \sum_{i=0}^N U_i(X; \theta, h; w) e^{-rt_i} P(X_i; \theta, h)$$

where $U_i(X; w)$ is the probability that the agent stops for the first time at t_i . Hence, U_i can be written as

$$U_i(X; \theta, h; w) = (1 - u_0(X_0; \theta, h; w)) \times \dots \times (1 - u_{i-1}(X_{i-1}; \theta, h; w)) u_i(X_i; \theta, h; w).$$

We now explain the training process of the ANN. The goal is to maximize objective function J with respect to w . There are three difficulties in this optimization problem. The first difficulty is that J cannot be directly calculated since it is the expected value of a random quantity. To overcome this difficulty, we use a Monte Carlo approach in which we generate independent simulations of the process X_t and then take the sample mean. In other words, if $X^{(1)}, \dots, X^{(m)}$ are independent simulations of X , then we approximate J by

$$\hat{J}(\theta, h; w) = \frac{1}{m} (\tilde{J}(X^{(1)}; \theta, h; w) + \dots + \tilde{J}(X^{(m)}; \theta, h; w)).$$

The second difficulty is that we are not given the input data θ and h to train our ANN by them. To overcome this difficulty, the algorithm itself has to generate the input data. This is accomplished by generating random values of θ and h as input data. The distribution of the random numbers are chosen so that the support of the distribution covers the reasonable range of the parameters.

The third and most subtle difficulty is a technical problem arising during the training phase of the ANN. Indeed, if one tries to simultaneously train all u_i s, it will be observed that, as i becomes larger and closer to N , the weights in the corresponding u_i are very slowly or not at all changing, and hence the ANN is not trained properly. The cause of this problem is the fact that when the sample paths are stopped in an earlier time t_j , they no longer participate in the payoff calculation for later t_i s. This reduces the sample size of the simulation and makes the training slow. To overcome this difficulty, we have used a novel approach for the training of our ANN. We notice that we can optimize the weights of each

u_i by only considering the portion of sample paths beyond time t_i . In other words, if we define

$$J_i(\theta, h; w) = \mathbb{E}[\tilde{J}_i(X_{i:}; \theta, h; w)],$$

where $X_{i:}$ denotes the restriction of X_t to $t \geq t_i$, and \tilde{J}_i is the payoff of the agent starting from time t_i and following the sample path $X_{i:}$, i.e.,

$$\tilde{J}_i(X; \theta, h; w) = \sum_{j=i}^N U_j(X; \theta, h; w) e^{-rt_j} P(X_j; \theta, h),$$

where we correspondingly define \hat{J}_i as the sample mean approximation of J_i .

Note that each \hat{J}_i only depends on later u_j s and not the earlier ones. This suggests that we can train the u_i in a backward manner. Hence, we start from $i = N$ and optimize \hat{J}_N , and then we go backward and optimize \hat{J}_i for $i = N - 1, \dots, 0$. After optimizing \hat{J}_0 , the whole ANN has been trained.

The maximization of each \hat{J}_i is a stochastic optimization problem since \hat{J}_i is indeed a noisy objective function. A successful method in such a setting is the Adam optimization method [27], which is a gradient-based method. The Adam method is a gradient ascent method and at each step calculates the gradient of the objective function and updates the weights by a step size (learning rate). The gradients are efficiently calculated by the back-propagation algorithm. We have chosen the step size to decay with an exponential rate of 0.9.

To initialize the weights and biases of ANN we use the uniform Xavier initialization [28]. The property of this initialization is that the variance of activations are the same across each layer.

To conclude this section, we present the complete algorithm (Algorithm 1) of our method here.

Algorithm 1 ANN for optimal stopping.

Input: Parameters of the SDE and payoff function

Output: Optimal weights of stopping probability functions

- 1: Initialize the weights of the ANN
 - 2: **for** $i = N, \dots, 0$ **do**
 - 3: **for** each epoch **do**
 - 4: Generate a batch of θ, h values by sampling from random uniform distributions
 - 5: Feed θ, h s into the ANN and obtain the functions $u_i(x; \theta, h; w)$
 - 6: Generate m independent simulations $X^{(1)}, \dots, X^{(m)}$ of the SDE (1)
 - 7: Evaluate u_i s on each $X^{(j)}$ and calculate the payoff for each simulation
 - 8: Average over sample payoffs to obtain the expected payoff \hat{J}_i
 - 9: Obtain gradients of \hat{J}_i w.r.t weights using back-propagation
 - 10: Update the weights of u_i by the Adam optimizer with given step size
 - 11: Adjust the step size by an exponential decay of given rate
 - 12: **Return** w
-

4. Application

In this section, we illustrate the method of the previous section in a problem arising from commodity optimal decision making in relation to changes in parameters and shocks. We assume that the demand for a commodity follows a stochastic differential equation such as Equation (1) with parameters θ . Later in this section, we will provide two explicit examples for the drift coefficient μ and volatility coefficient σ and will illustrate the result of our method for them.

The commodity price is then calculated by applying the demand function to the demand process. We use the well-known linear demand function $P(x) = (b - ax)^+$. In this setting, the optimal stopping problem can be viewed as a model for the optimal storage

problem. Indeed, if we assume that an agent has the possibility of storing the commodity in order to sell it in a future time, then the agent will choose the optimal time τ to sell so that the quantity $e^{-r\tau}P(X_\tau)$ is maximized. Having access to the storage option is very important to cope with the shocks to the demand fundamental parameters or even the demand value itself at the time of common shocks such as COVID-19 economic downturns.

Hence, for such an agent, the value of the storage option will be

$$V(t, x) = \sup_{t \leq \tau \leq T} \mathbb{E}[e^{-r\tau}P(X_\tau)|X_t = x],$$

where x is the current demand in the market.

We are now ready to train an ANN for this optimal stopping problem. The input of the ANN will be the model parameters $\Theta = (\theta, a, b, r)$ and also the impulse h . θ indicates the first set of parameters, which govern the SDE, and the second set of parameters consists of a, b, r , which determine the payoff. The simulations will be carried out using the Euler–Maruyama method for SDEs (see [29]). To generate inputs for the ANN, we use independent uniform random samples for each of the parameters. The range of the uniform distributions is chosen so that it covers the estimated values for real data obtained in [19]. We then train the ANN and obtain the optimal weights w^* , which determine the optimal stopping probabilities $u_i(\theta, h; w^*)$. Unlike most applications of ANN, instead of minimizing an error or loss, here we maximize the expected payoff. However, we can still use a learning curve to illustrate the improvement of learning. We have to mention that the total simulation sample size is 16,000.

The crucial question from the agent standpoint is that of when to sell the inventory. In other words, what is the optimal stopping rule? As mentioned previously, there exists a stopping region D where the agent should stop when the demand enters this region. The boundary of the stopping region, known as the stopping boundary, may have complicated shapes in general. We assume that the stopping boundary is simply a single point x^* and the stopping region becomes the interval (x^*, ∞) . The authors of [19] have assumed x^* to be a parameter and then estimated x^* along other parameters using statistical methods.

In our setting, the stopping occurs at time t_i by a probability $u_i(x; \theta, h; w^*)$. To convert the stopping probability to a stopping region, we fix a confidence level $\gamma \in (0, 1)$ and take the stopping region to be the points in which the stopping occurs with a probability greater than γ . We can also define the stopping boundary x^* as

$$x_i^*(\theta) = \inf\{x : u_i(x; \theta, h; w^*) > \gamma\}.$$

In the rest of this section, we introduce two specific models for the demand process and then illustrate the result of our method on the two models.

4.1. Exponential Ornstein-Uhlenbeck Model

Our models for commodity demand should satisfy two criteria. One is that the demand should always be non-negative, and the other is the mean-reverting nature of the commodities. The two models that we introduce in this section both satisfy these two criteria.

We use the Exponential Ornstein–Uhlenbeck Model (EOU) [19,30]. We assume that the log-demand for a commodity is governed by the following SDE:

$$dZ_t = \alpha(\mu - Z_t)dt + \sigma dB_t,$$

where μ is the long-term average demand and σ is the volatility of demand and α is the mean reversion rate. The demand process then becomes $X_t = (1 + h)Y_t$, where $Y_t = e^{Z_t}$. Although the process X_t is also governed by a SDE, but we do not need it because we can simulate X_t by first simulating Z_t and then calculating $(1 + h)e^{Z_t}$.

4.2. Cox–Ingersoll–Ross Model

The Cox–Ingersoll–Ross (CIR) process follows the following SDE:

$$dY_t = \alpha(\mu - Y_t)dt + \sigma\sqrt{Y_t}dB_t.$$

Let $X_t = (1 + h)Y_t$. This model is used for commodity consumption modeling, see [18]. The parameters of this equation has the same interpretation as the Ornstein–Uhlenbeck process except that here the diffusion term is proportional to $\sqrt{X_t}$. It can be shown that this equation has a unique non-negative solution ([31], Theorem 6.2.2). Moreover, under the assumption $2\alpha\mu \geq \sigma^2$, X_t is always positive.

4.3. Numerical Assessment

Now, we focus on the results of the numerical assessment. However, before that, let us review the changes that the economies has experienced during the COVID-19 crisis. Here, we present a chart that includes the “all spending” as well as the “grocery spending” over the past two years in the USA. The data is taken from <https://tracktherecovery.org/> accessed on 30 December 2022 provided by Affinity Solutions. As one can see in Figure 2, while the total spending has sharply reduced, which can indicate the reduction in most of the goods during the COVID-19 crisis, groceries have witnessed a sharp increase. For that reason, we need to examine the negative and positive changes in the demand parameters as well as the shock process, which give rise to sensitivity and impulse analysis. This is what we are going to present in the coming discussions.

The goal of the manuscript is to provide a numerical method for the sensitivity analysis of a stochastic optimal stopping problem. The data are generated from stochastic processes and then used to train the artificial neural network. The calibration to the real data is a difficult task on its own (see for instance [19]), that can be the subject of a future work.

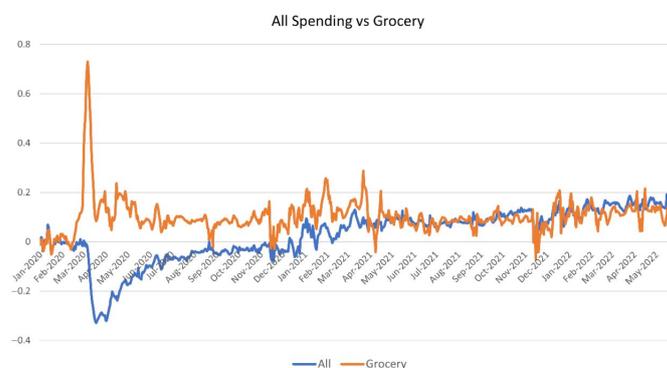


Figure 2. The USA consumption over the past two years.

As mentioned earlier, we have trained an ANN to learn about the optimal stopping problem for a set of fixed parameters, to see what the learning curves would look like. We use a default parameter set. The default parameters are $\alpha = 0.5$, $\mu = 0.2$, $\sigma = 1$, $r = 0.1$, $a = 1$, $b = 2$ and $h = 0$. Later, the parameters will also be part of the input data.

To validate our method, we consider the exponential OU process and find the value function to make a comparison with the results in [19] that uses the same model, but with a different method. Figure 3 shows how the expected payoff behaves during the training process. The x axis shows the number of epochs. As can be seen in the figure, the payoff reaches its maximum after around 250 epochs. After that point, it shows minor fluctuations, which are due to the inherent randomness in the calculation of the expected payoff.

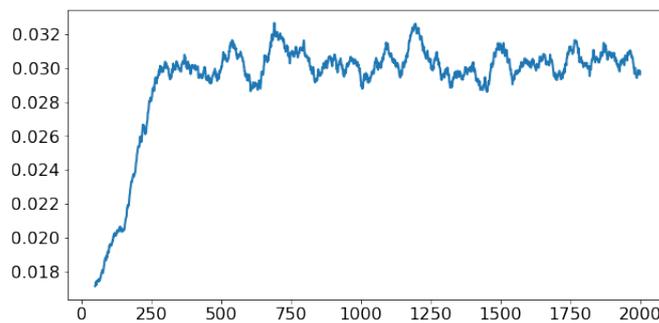


Figure 3. The expected payoff as a function of the number of epochs, evaluated for the parameters $\alpha = 0.5, \mu = 0.2, \sigma = 1, r = 0.1, a = 1, b = 2$ and $h = 0$.

Furthermore, as depicted in Figure 3, the learning curve stabilizes after 250 iterations, which is a sign of successful model validation in a neural network.

Figure 4 plots the value function $V(x)$ for the different values of each of its parameters. The figures reveal much information about the value function. First, as can be seen, $V(x)$ is a convex and decreasing function of x for all the illustrated values of parameters.

Second, the beginning segment of the graph of V is a linear segment. The reason for this is that for $x \in (0, x_0^*)$ we have $V(x) = P(x)$. This justifies our assumption that the domain D is in the form of $(0, x_0^*)$.

Third, changing each parameter affects the value function in a monotonic way, so that V seems to be increasing with respect to σ and b and decreasing with respect to the remaining parameters. We will see more about the monotonicity in the sensitivity analysis section.

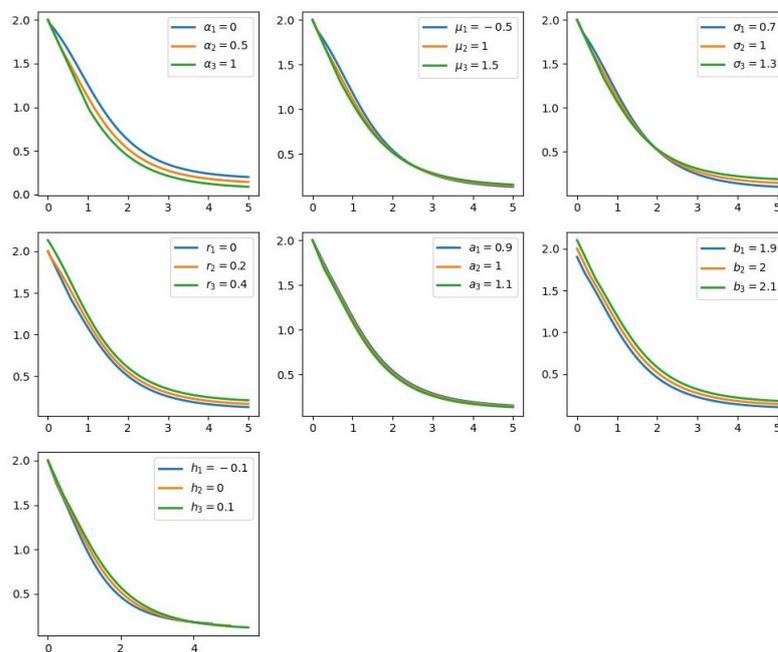


Figure 4. Each figure shows the graph of $V(x)$ for three different values of the indicated parameter. The fixed parameters in each graph are chosen as $\alpha = 0.5, \mu = 0.2, \sigma = 1, r = 0.1, a = 1, b = 2$.

Comparing to the results produced in [19], the simulations match with our expectations; for instance, we see how the prices evolve with the volatility as higher volatility yields higher prices. However, we leave the interpretation of the other parameters to be read from [19]. Overall, these results show that the learning process of the neural network has generated a set of rightful value functions.

4.4. Sensitivity and Impulse Analysis

Now, let us focus on sensitivity and impulse analysis. Sensitivity analysis is essentially to see how the model reacts to changes in the model parameters, and impulse means how the model reacts to a shock to the demand. The shock to the demand is implemented by multiplying the demand process by $1 + h$, for a small parameter h , which represents the percentage changes in the demand.

Sensitivity and impulse analysis can be useful for a number of reasons, such as estimating how the time for the optimal storage time would delay or accelerate due to the economic shocks such as the COVID-19 catastrophe. The proposed sensitivity analysis method in this paper has applications beyond just COVID-19, as it can be used in a variety of contexts. Particularly, they can be used for calculating the Greeks of an American option or creating neutral portfolios to hedge against various risks. For example, delta hedging is a common strategy used to create a delta neutral portfolio, which is not sensitive to small changes in the underlying parameter.

4.4.1. Optimal Storage

First, let us focus on the optimal storage value x^* . The Figures 5 and 6 plot u_1 as a function of x for different values of parameters $\alpha, \mu, \sigma, r, a, b$ and also impulse h , for the demand process exponential OU and CIR, respectively. The confidence level $\gamma = 0.95$ is illustrated in the figures. The corresponding value of x^* can be obtained by intersecting the confidence line with each graph.

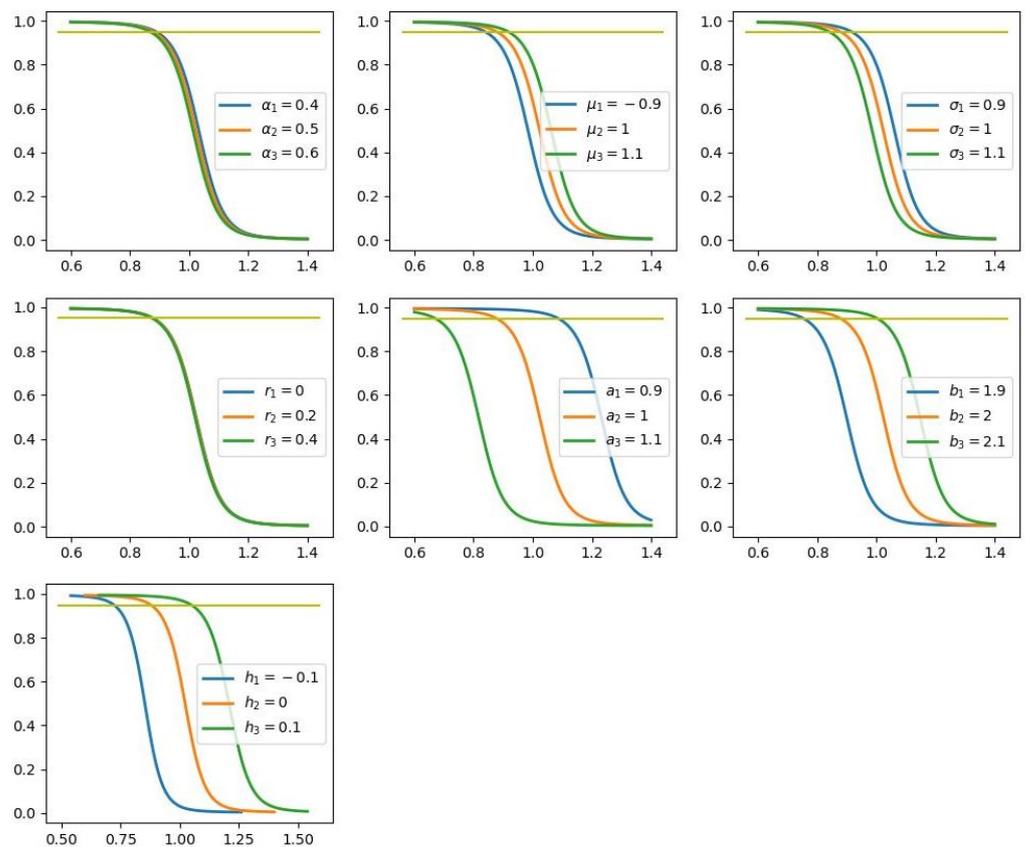


Figure 5. The figures show the graph of $u_1(x)$ for three different values of each parameter indicated in the figures for the exponential OU model. The horizontal line cuts the probability of stop at a confidence level of 95%.

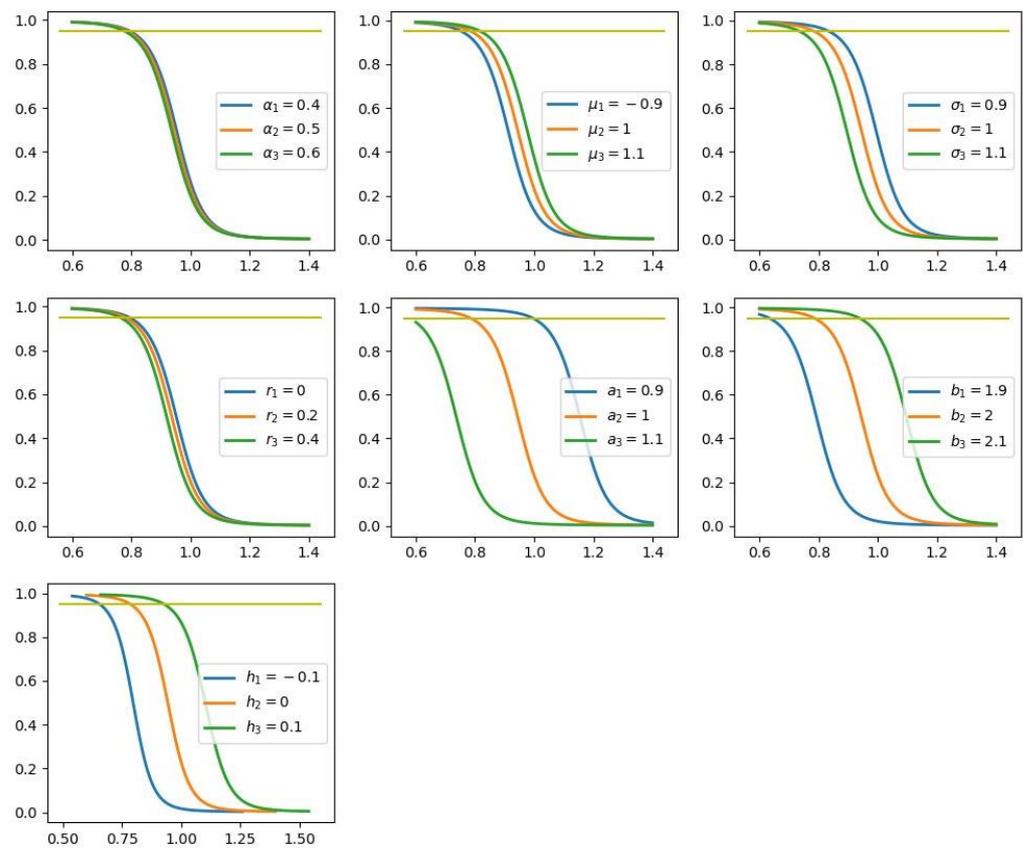


Figure 6. The figures show the graph of $u_1(x)$ for three different values of each parameter indicated in the figures for the CIR model. The horizontal line cuts the probability of stop at a confidence level of 95%.

Interestingly, the results for the two models are rather consistent, which shows the optimal storage x^* is to a good extent to the choice of model. As one can see that the results, however, vary for different parameters. For α , x^* is decreasing, which indicates the larger mean reversion rate is one at a smaller value of the good it would decide to store. On the other hand, for μ , the relation is the opposite, meaning that the larger the average consumption, the larger storage value would be. Particularly, in the COVID-19 case, we can expect to see smaller μ for overall consumption and larger for the grocery. The behavior for σ is in a reverse direction than for μ , as more volatile demand would imply a decision for storage at a smaller value of the commodity. This is particularly important in COVID-19, as the demands for many goods have become volatile. The same is true for r and a . Note that increasing r means a higher time value, and larger a means more inelastic demand. The changes for the parameter b are not so meaningful given that we can always assume it is equal to 1. However, as one can see, the optimal value to store with respect to impulse h is increasing. This means in positive shocks there is less fear for price drops, so there is no need to rush storing goods, which makes sense.

4.4.2. Value Functions

Once the optimal decision problem has been solved, it is important to understand how the solution would react to changes in the model and impulse parameters.

One of the challenges of sensitivity and impulse analysis in the optimal stopping problem is the complex dependence of the optimal value function on the model parameters. One way to address this is to approximate the optimal value function with a simple parametric form and analyze the impulsive and sensitivity of the approximation, see [19]. However, simple parametric forms may not capture the complexity of the value function

as a function of the parameters, and increasing the complexity of the parametric form can make it difficult to optimize the approximation. A network-based approach is able to capture the complexity of the value function and has well-developed optimization methods for training. This makes it a useful tool for impulse and sensitivity analysis in the optimal stopping problem.

We now focus on the exponential OU and CIR example of the previous section. We conduct a sensitivity analysis for the value function.

The back-propagation enables us to calculate efficiently the partial derivatives of V with respect to its parameters. This gives us a second order sensitivity analysis and illustrates the convexity and concavity with respect to each parameter. The figures of the partial derivative are shown in Figures 7 and 8. As can be seen, the signs of the partial derivatives justify the monotonicity behavior that we observed in the graph of V previously.

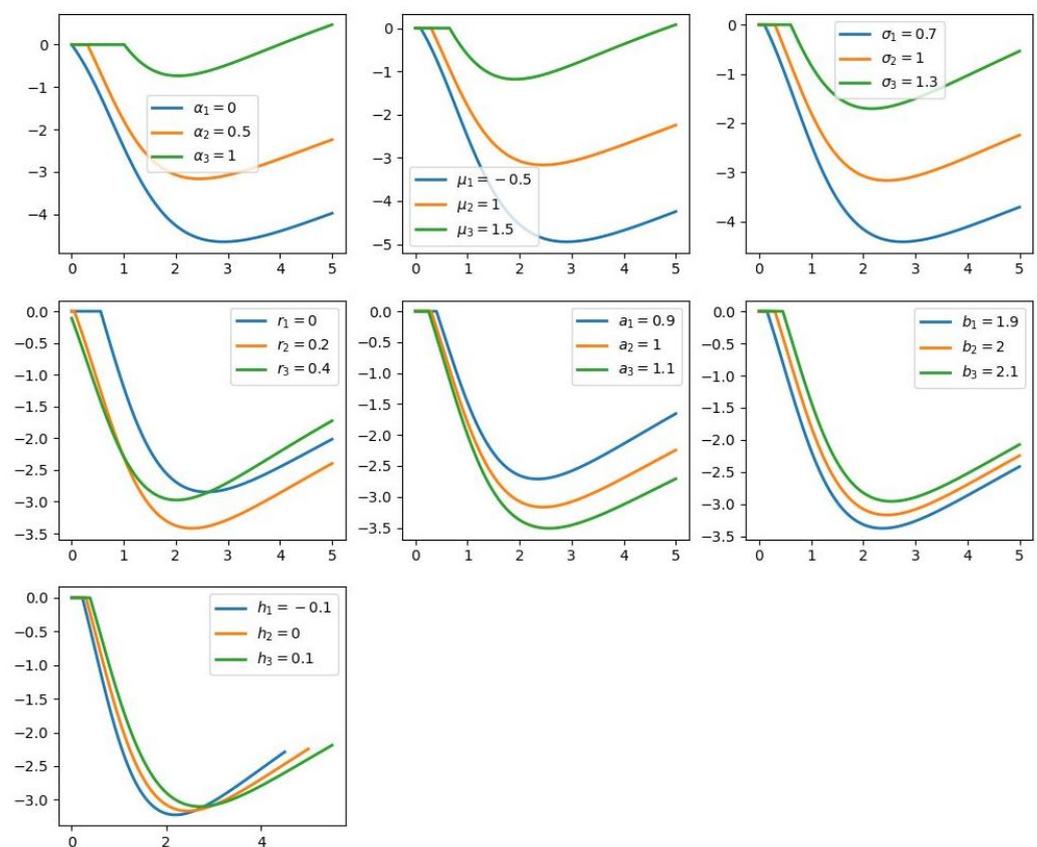


Figure 7. Sensitivity and impulse analysis of the value function of the exponential OU model. Each figure shows the graph of the partial derivative of $V(x)$ with respect to one of its parameters, for three different values of the parameter. The fixed parameters in each graph are chosen as $\alpha = 0.5, \mu = 0.2, \sigma = 1, r = 0.1, a = 1, b = 2, h = 0$.

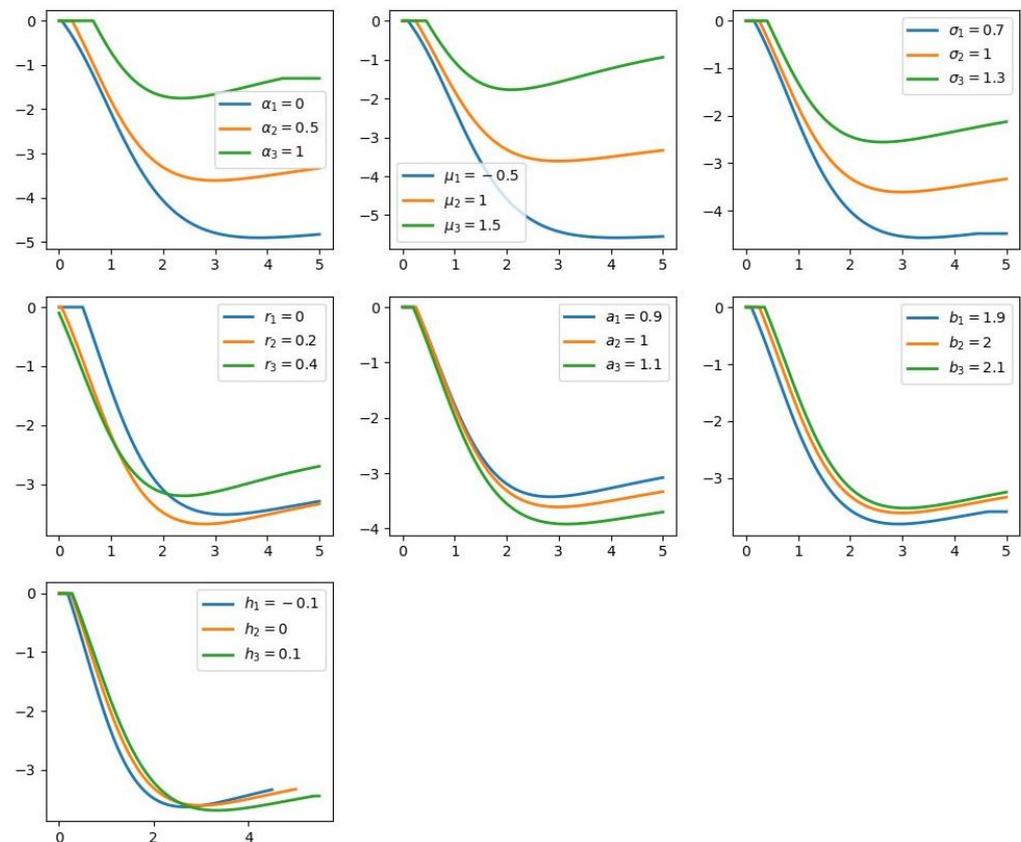


Figure 8. Sensitivity and impulse analysis of the value function of the CIR model. Each figure shows the graph of the partial derivative of $V(x)$ with respect to one of its parameters, for three different values of the parameter. The fixed parameters in each graph are chosen as $\alpha = 0.5, \mu = 0.2, \sigma = 1, r = 0.1, a = 1, b = 2, h = 0$.

First of all, as one can observe as with the optimal storage value, there is a consistency between the results of the two models, which makes the results very robust and reliable with respect to the changes in the modeling.

Second, the first part of all figures is constantly zero. The reason is that, for $x < x^*$, the value function coincided with $b - ax$, and hence is independent of all parameters except a .

Third, as we expect, the first part of all figures are compatible with the value of x^* that we have obtained in the previous part for optimal storage, as larger x^* means larger values at which the figures remain equal to zero. However, in the continuation, the patterns do not always follow the same uniformity, with some figures exceeding others. Interestingly, there are values of x where in almost all figures the partial derivative reaches a minimum. This means that at a particular value, if a parameter changes, it is important whether the change is positive or negative. This means that the sensitivity of the value function to changes in the parameters and the impulse at a particular point is the smallest.

Forth, as from Figure 4, we can see that naturally if x^* is larger, the value function itself is smaller. However, as one can see from Figures 7 and 8, the opposite is true for the demand model parameters. This means at any value of the parameters, for larger x , the impact of the changes in those parameters are also larger. Interestingly, for the impulse, the behavior is changing at some point, which means while larger values for h gives larger partial derivatives with respect to h in the beginning, in one point they change the order.

As a final remark, we should mention that one can easily conduct a similar sensitivity analysis for the optimal stopping boundary, x^* . Indeed, since each x_i^* is obtained from u_i by a simple operation, one can still use back-propagation to calculate its partial derivatives.

Remark 1. A key benefit of our approach is that it does not depend on an analytical expression of the model and only necessitates simulated samples from the model. As a result, it can be utilized even in jump-diffusion models, although validation must be conducted for this new model. In other words, our approach is model-free.

5. Concluding Remarks

This paper presents a new method for the optimal stopping problem using artificial neural networks, that can be applied to situations where demand is shocked such as during the COVID-19 pandemic. As previously mentioned, the COVID-19 pandemic resulted in a substantial shock to the demand for certain goods, particularly food. This underscores the need for robust methods to analyze the implications of such shocks in existing economic and financial models. In this article, we concentrated on examining the impact of a shock on the price of a storable commodity. When agents are permitted to store the commodity for potential future sales, they must solve an optimal stopping problem. Thus, the value of the commodity is derived from the optimal value function. One advantage of this approach is its ability to efficiently conduct a sensitivity analysis. The method is demonstrated through two different models for commodity demand processes, where the value function and optimal stopping boundaries are calculated and sensitivity analysis is performed on all parameters, including volatility and interest rate. We obtain very robust results even after we change the models. While the optimal storage value is very consistent over all different changes, the changes to value function have different behaviors, which makes interpretation much harder. However, the results are very useful for the quantitative analysis due to robustness. The proposed method has potential applications in a wide range of optimal stopping problems in various fields, including finance.

Let us discuss some limitations and future directions of this study. A technical constraint is that, although the method is model-free, the convergence of the learning curve may depend on the nature of the process. In our case, both models, the exponential OU and CIR, are mean-reverting to a good extent, which aids in achieving convergence. However, if we were to use Geometric Brownian Motions, which are not mean-reverting, we cannot ensure that convergence will occur in a finite amount of time.

Another limitation is that the storage model considered in this study does not apply to all commodities. In particular, precious metals do not follow the same storage rules, and the model works best for agricultural goods.

There are also several interesting extensions of this model to consider in future studies. One can address the issues mentioned above, and additionally, observe the model's performance on real data. This is a crucial task that requires a separate study, such as the one in [19], which illustrates the difficulties that arise when dealing with real data. Another future direction is to use the same methodology for non-commodity assets, such as using our method to determine the Greeks of American options.

Author Contributions: Writing—original draft, N.K., E.S. and H.A. (Hirbod Assa); Supervision, H.A. (Hojatollah Adibi). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Başaran, B.; Pekmezci Purut, H. The impact of the COVID-19 pandemic on the frequency of food consumption. *J. Tour. Gastron. Stud.* **2021**, *9*, 47–66. [[CrossRef](#)]
2. Coulthard, H.; Sharps, M.; Cunliffe, L.; van den Tol, A. Eating in the lockdown during the COVID 19 pandemic; Self-reported changes in eating behaviour, and associations with BMI, eating style, coping and health anxiety. *Appetite* **2021**, *161*, 105082. [[CrossRef](#)]

3. Fiorella, K.J.; Bageant, E.R.; Mojica, L.; Obuya, J.A.; Ochieng, J.; Olela, P.; Otuo, P.W.; Onyango, H.O.; Aura, C.M.; Okronipa, H. Small-scale fishing households facing COVID-19: The case of Lake Victoria, Kenya. *Fish. Res.* **2021**, *237*, 105856. [[CrossRef](#)]
4. Pabst, A.; Bollen, Z.; Creupelandt, C.; Fontesse, S.; Orban, T.; de Duve, M.; Pinon, N.; Maurage, P. Alcohol consumption changes during the first COVID-19 lockdown: An online population survey in a convenience sample of French-speaking Belgian residents. *Psychiatry Res.* **2021**, *300*, 113938. [[CrossRef](#)]
5. Güngör, B.O.; Ertuğrul, H.M.; Soytaş, U. Impact of COVID-19 outbreak on Turkish gasoline consumption. *Technol. Forecast. Soc. Chang.* **2021**, *166*, 120637. [[CrossRef](#)]
6. Xiong, J.; Tang, Z.; Zhu, Y.; Xu, K.; Yin, Y.; Xi, Y. Change of Consumption Behaviours in the Pandemic of COVID-19: Examining Residents' Consumption Expenditure and Driving Determinants. *Int. J. Environ. Res. Public Health* **2021**, *18*, 9209. [[CrossRef](#)]
7. Gustafson, R.L. Implications of recent research on optimal storage rules. *J. Farm Econ.* **1958**, *40*, 290–300. [[CrossRef](#)]
8. Muth, J.F. Rational expectations and the theory of price movements. *Econometrica* **1961**, *29*, 315–335. [[CrossRef](#)]
9. Samuelson, P.A. Stochastic Speculative Price. *Proc. Natl. Acad. Sci. USA* **1971**, *68*, 335–337. [[CrossRef](#)]
10. Deaton, A.; Laroque, G. On the behaviour of commodity prices. *Rev. Econ. Stud.* **1992**, *59*, 1–23. [[CrossRef](#)]
11. Deaton, A.; Laroque, G. Estimating a nonlinear rational expectations commodity price model with unobservable state variables. *J. Appl. Econom.* **1995**, *10*, 9–40. [[CrossRef](#)]
12. Deaton, A.; Laroque, G. Competitive storage and commodity price dynamics. *J. Political Econ.* **1996**, *104*, 896–923. [[CrossRef](#)]
13. Chambers, M.J.; Bailey, R.E. A theory of commodity price fluctuations. *J. Political Econ.* **2001**, *104*, 924–957. [[CrossRef](#)]
14. Ng, S.; Ruge-Murcia, F.J. Explaining the persistence of commodity prices. *Comput. Econ.* **2000**, *16*, 149–171. [[CrossRef](#)]
15. Cafiero, C.; Bobenrieth, E.S.; Bobenrieth, J.R.; Wright, B. The empirical relevance of the competitive storage model. *J. Econom.* **2011**, *162*, 44–54. [[CrossRef](#)]
16. Miao, Y.W.W.; Funke, N. *Reviving the Competitive Storage Model; A Holistic Approach to Food Commodity Prices*; IMF Working Paper; International Monetary Fund: Washington, DC, USA, 2011.
17. Arseneau, D.; Leduc, S. Commodity price movements in a general equilibrium model of storage. *IMF Econ. Rev.* **2012**, *61*, 199–224. [[CrossRef](#)]
18. Assa, H. A financial engineering approach to pricing agricultural insurances. *Agric. Financ. Rev.* **2015**, *75*, 63–76. [[CrossRef](#)]
19. Karimi, N.; Assa, H.; Salavati, E.; Adibi, H. Calibration of Storage Model by Multi-Stage Statistical and Machine Learning Methods. *Comput. Econ.* **2022**, 1–19. [[CrossRef](#)]
20. Becker, S.; Cheridito, P.; Jentzen, A. Deep optimal stopping. *J. Mach. Learn. Res.* **2019**, *20*, 74.
21. Becker, S.; Cheridito, P.; Jentzen, A. Pricing and hedging American-style options with deep learning. *J. Risk Financ. Manag.* **2020**, *13*, 158. [[CrossRef](#)]
22. Burlacu, C.M.; Burlacu, A.C.; Praisler, M. Sensitivity Analysis of Artificial Neural Networks Identifying JWH Synthetic Cannabinoids Built with Alternative Training Strategies and Methods. *Inventions* **2022**, *7*, 82. [[CrossRef](#)]
23. Guerra, M.; Nunes, C.; Oliveira, C. The optimal stopping problem revisited. *Stat. Pap.* **2021**, *62*, 137–169. [[CrossRef](#)]
24. Jelito, D.; Pitera, M.; Stettner, L. Risk sensitive optimal stopping. *Stoch. Process. Their Appl.* **2021**, *136*, 125–144. [[CrossRef](#)]
25. Herrera, C.; Krach, F.; Ruysen, P.; Teichmann, J. Optimal stopping via randomized neural networks. *arXiv* **2021**, arXiv:2104.13669.
26. Becker, S.; Cheridito, P.; Jentzen, A.; Welti, T. Solving high-dimensional optimal stopping problems using deep learning. *Eur. J. Appl. Math.* **2021**, *32*, 470–514. [[CrossRef](#)]
27. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
28. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
29. Kloeden, P.E.; Pearson, R.A. The numerical solution of stochastic differential equations. *J. Aust. Math. Soc. Ser. B Appl. Math.* **1977**, *20*, 8–12. [[CrossRef](#)]
30. Schwartz, E.S. The stochastic behavior of commodity prices: Implications for valuation and hedging. *J. Financ.* **1997**, *52*, 923–973. [[CrossRef](#)]
31. Lamberton, D.; Lapeyre, B. *Introduction to Stochastic Calculus Applied to Finance*; CRC Press: Boca Raton, FL, USA, 2011.

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.