ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Deep optimal stopping

S. Becker and P. Cheridito and A. Jentzen

Research Report No. 2018-14
April 2018

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Deep optimal stopping*

Sebastian Becker[†] Patrick Cheridito[‡] & Arnulf Jentzen[§]

April 2018

Abstract

In this paper we develop a deep learning method for optimal stopping problems which directly learns the optimal stopping rule from Monte Carlo samples. As such it is broadly applicable in situations where the underlying randomness can efficiently be simulated. We test the approach on two benchmark problems: the pricing of a Bermudan max-call option on different underlying assets and the problem of optimally stopping a fractional Brownian motion. In both cases it produces very accurate results in high-dimensional situations with short computing times.

Keywords: optimal stopping, deep learning, Bermudan option, fractional Brownian motion

1 Introduction

We consider optimal stopping problems of the form $\sup_\tau \mathbb{E}\, g(\tau, X_\tau)$, where $X = (X_n)_{n=0}^N$ is an $\mathbb{R}^d$-valued discrete-time Markov process and the supremum is over all stopping times τ based on observations of X. Formally, this just covers situations where the stopping decision can only be made at finitely many times. But practically all relevant continuous-time stopping problems can be approximated with time-discretized versions. The Markov assumption means no loss of generality. We make it because it simplifies the presentation and many important problems already are in Markovian form. But every optimal stopping problem can be made Markov by including all relevant information from the past in the current state of X (albeit at the cost of increasing the dimension of the problem).

In theory, optimal stopping problems with finitely many stopping opportunities can be solved exactly. The optimal value is given by the smallest supermartingale that dominates the reward process – the so-called Snell envelope – and the smallest (largest) optimal stopping time is the

*We thank Ariel Neufeld and Martin Stefanik for fruitful discussions and helpful comments.

[†]Mathematics Institute, Goethe University Frankfurt, 60325 Frankfurt am Main, Germany and Zenai AG, 8045 Zurich, Switzerland; sebastian.becker@zenai.ch

[‡]RiskLab, Department of Mathematics, ETH Zurich, 8092 Zurich, Switzerland; patrick.cheridito@math.ethz.ch

[§]SAM, Department of Mathematics, ETH Zurich, 8092 Zurich, Switzerland; arnulf.jentzen@sam.math.ethz.ch

first time when the immediate reward dominates (exceeds) the continuation value; see, e.g., [33, 38]. But traditional numerical methods suffer from the curse of dimensionality. For instance, the complexity of standard tree- and lattice-based methods increases exponentially in the dimension. For typical problems they yield good results up to three dimensions. To treat higher dimensional problems, various Monte Carlo based methods have been developed over the last years. For different approaches to approximate the Snell envelope or continuation value, we refer to [36, 44, 11, 12, 4, 30, 19, 9, 29, 25] and the references therein. A different strand of the literature has focused on computing the exercise boundary; see, e.g., [2, 20, 5]. Based on an idea of [15], a dual approach was developed by [39, 24]; see [27, 13] for a multiplicative version and [3, 10, 7, 40, 17, 6, 8, 34] for extensions and primal-dual methods.

In this paper we use deep learning to directly approximate an optimal stopping time. While conceptually, our approach is simpler than most of the existing methods, its challenge lies in the implementation of a deep learning method that can well approximate stopping times. We do this by decomposing an optimal stopping time into a sequence of 0-1 stopping decisions and learning them recursively with a multilayer feedforward neural network using backpropagation together with a stochastic gradient ascent optimization algorithm. Since our method produces a candidate optimal stopping time, it leads to a low-biased estimate of the optimal value $\sup_\tau \mathbb{E}\, g(\tau, X_\tau)$. However, in examples for which there exist reference solutions in the literature, it yields highly accurate results with short computing times.

The rest of the paper is organized as follows: In Section 2 we introduce the setup and explain our method of approximating optimal stopping times by multilayer feedforward neural networks. In Section 3 we test the approach on two benchmark examples: the pricing of a Bermudan max-call option on different underlying assets and the problem of optimally stopping a fractional Brownian motion. In the first case we use a multi-dimensional Black–Scholes model to describe the dynamics of the underlying assets. Then the pricing of a Bermudan max-call option amounts to solving a d-dimensional optimal stopping problem, where d is the number of assets. We provide numerical results for $d = 5, 10, 20, 30, 50, 100, 200$ and 500. In the second example we only consider a one-dimensional fractional Brownian motion. But fractional Brownian motion is not a Markov process. In fact, all of its increments are correlated. So to optimally stop it, one has to keep track of all past movements. To make it tractable, we approximate the continuous-time problem with a time-discretized version, which if formulated as a Markovian problem, has as many dimensions as there are time-steps. We compute a solution for 100 time-steps.

2 Deep learning optimal stopping rules

Let $X = (X_n)_{n=0}^N$ be an $\mathbb{R}^d$-valued discrete-time Markov process on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where N and d are positive integers. We denote by $\mathcal{F}_n$ the σ-algebra generated by $X_0, X_1, \ldots, X_n$ and call a random variable $\tau\colon \Omega \to \{0, 1, \ldots, N\}$ an X-stopping time if the event $\{\tau = n\}$ belongs to $\mathcal{F}_n$ for all $n \in \{0, 1, \ldots, N\}$.

Our aim is to develop a deep learning method that can efficiently learn an optimal policy for stopping problems of the form

$$\sup_{\tau\in\mathcal{T}} \mathbb{E}\, g(\tau, X_\tau), \tag{1}$$

where $g\colon \{0,1,\dots,N\}\times\mathbb{R}^d\to\mathbb{R}$ is a measurable function and $\mathcal{T}$ denotes the set of all X-stopping times. To make sure that problem (1) is well-defined and admits an optimal solution, we assume that g satisfies the integrability condition

$$\mathbb{E}\,|g(n,X_n)|<\infty \quad \text{for all } n\in\{0,1,\dots,N\} \tag{2}$$

(see, e.g., [33, 38]).

2.1 Expressing stopping times in terms of stopping decisions

Any X-stopping time can be decomposed into a sequence of 0-1 stopping decisions. In principle, the decision whether to stop the process at time n if it has not been stopped before, can be made based on the whole evolution of X from time 0 until n. But to optimally stop the Markov process X, it is enough to make stopping decisions according to $f_n(X_n)$ for measurable functions $f_n\colon\mathbb{R}^d\to\{0,1\}$, $n=0,1,\dots,N$. Theorem 1 below extends this well-known fact and serves as the theoretical basis of our method.

Consider the auxiliary stopping problems

$$V_n=\sup_{\tau\in\mathcal{T}_n}\mathbb{E}\, g(\tau,X_\tau) \tag{3}$$

for $n=0,1,\dots,N$, where $\mathcal{T}_n$ is the set of all X-stopping times satisfying $n\le\tau\le N$. Obviously, $\mathcal{T}_N$ consists of the unique element $\tau_N=N$, and one can write $\tau_N=Nf_N(X_N)$ for the constant function $f_N\equiv 1$. Moreover, for given $n\in\{0,1,\dots,N\}$ and a sequence of measurable functions $f_n,f_{n+1},\dots,f_N\colon\mathbb{R}^d\to\{0,1\}$ with $f_N\equiv 1$,

$$\tau_n=\sum_{k=n}^{N} k f_k(X_k)\prod_{j=n}^{k-1}(1-f_j(X_j)) \tag{4}$$

defines[1] a stopping time in $\mathcal{T}_n$. The following result shows that, for our method of recursively computing an approximate solution to the optimal stopping problem (1), it will be sufficient to consider stopping times of the form (4).

Theorem 1. *For a given $n\in\{0,1,\dots,N-1\}$, let τ_{n+1} be a stopping time in $\mathcal{T}_{n+1}$ of the form*

$$\tau_{n+1}=\sum_{k=n+1}^{N} k f_k(X_k)\prod_{j=n+1}^{k-1}(1-f_j(X_j)), \tag{5}$$

[1] In expressions of the form (4), we understand the empty product $\prod_{j=n}^{n-1}(1-f_j(X_j))$ as 1.

for measurable functions $f_{n+1}, \dots, f_N \colon \mathbb{R}^d \to \{0,1\}$ *with* $f_N \equiv 1$. *Then there exists a measurable function* $f_n \colon \mathbb{R}^d \to \{0,1\}$ *such that the stopping time* $\tau_n \in \mathcal{T}_n$ *given by* (4) *satisfies*

$$\mathbb{E}\, g(\tau_n, X_{\tau_n}) \geq V_n - \big(V_{n+1} - \mathbb{E}\, g(\tau_{n+1}, X_{\tau_{n+1}})\big)\,,$$

where V_n *and* V_{n+1} *are the optimal values defined in* (3).

Proof. Denote $\varepsilon = V_{n+1} - \mathbb{E}\, g(\tau_{n+1}, X_{\tau_{n+1}})$, and fix a stopping time $\tau \in \mathcal{T}_n$. By the Doob–Dynkin lemma (see, e.g., Theorem 4.41 in [1]), there exists a measurable function $h_n \colon \mathbb{R}^d \to \mathbb{R}$ such that $h_n(X_n)$ is a version of the conditional expectation $\mathbb{E}\big[g(\tau_{n+1}, X_{\tau_{n+1}}) \mid X_n\big]$. Moreover, due to the special form (5) of τ_{n+1},

$$g(\tau_{n+1}, X_{\tau_{n+1}}) = \sum_{k=n+1}^{N} g(k, X_k) I_{\{\tau_{n+1}=k\}} = \sum_{k=n+1}^{N} g(k, X_k) I_{\left\{f_k(X_k)\prod_{j=n+1}^{k-1}(1-f_j(X_j))=1\right\}}$$

is a measurable function of $X_{n+1}, \dots, X_N$. So it follows from the Markov property of X that $h_n(X_n)$ is also a version of the conditional expectation $\mathbb{E}\big[g(\tau_{n+1}, X_{\tau_{n+1}}) \mid \mathcal{F}_n\big]$. Since the events

$$D = \{g(n, X_n) \geq h_n(X_n)\} \quad \text{and} \quad E = \{\tau = n\}$$

are in $\mathcal{F}_n$, $\tau_n = nI_D + \tau_{n+1} I_{D^c}$ belongs to $\mathcal{T}_n$ and $\tilde{\tau} = \tau_{n+1} I_E + \tau I_{E^c}$ to $\mathcal{T}_{n+1}$. It follows from the definitions of V_{n+1} and ε that $\mathbb{E}\, g(\tau_{n+1}, X_{\tau_{n+1}}) = V_{n+1} - \varepsilon \geq \mathbb{E}\, g(\tilde{\tau}, X_{\tilde{\tau}}) - \varepsilon$. This implies

$$\mathbb{E}\big[g(\tau_{n+1}, X_{\tau_{n+1}}) I_{E^c}\big] \geq \mathbb{E}[g(\tilde{\tau}, X_{\tilde{\tau}}) I_{E^c}] - \varepsilon = \mathbb{E}[g(\tau, X_\tau) I_{E^c}] - \varepsilon,$$

from which one obtains

$$\begin{aligned}
&\mathbb{E}\, g(\tau_n, X_{\tau_n}) = \mathbb{E}\big[g(n, X_n) I_D + g(\tau_{n+1}, X_{\tau_{n+1}}) I_{D^c}\big] = \mathbb{E}[g(n, X_n) I_D + h_n(X_n) I_{D^c}] \\
&\geq \mathbb{E}[g(n, X_n) I_E + h_n(X_n) I_{E^c}] = \mathbb{E}\big[g(n, X_n) I_E + g(\tau_{n+1}, X_{\tau_{n+1}}) I_{E^c}\big] \\
&\geq \mathbb{E}[g(n, X_n) I_E + g(\tau, X_\tau) I_{E^c}] - \varepsilon = \mathbb{E}\, g(\tau, X_\tau) - \varepsilon.
\end{aligned}$$

Since $\tau \in \mathcal{T}_n$ was arbitrary, this shows that $\mathbb{E}\, g(\tau_n, X_{\tau_n}) \geq V_n - \varepsilon$.

Now, note that if one defines $f_n \colon \mathbb{R}^d \to \{0,1\}$ by

$$f_n(x) = \begin{cases} 1 & \text{if } g(n,x) \geq h_n(x) \\ 0 & \text{if } g(n,x) < h_n(x) \end{cases},$$

one has $I_D = f_n(X_n)$. Therefore,

$$\tau_n = n f_n(X_n) + \tau_{n+1}(1 - f_n(X_n)) = \sum_{k=n}^{N} k f_k(X_k) \prod_{j=n}^{k-1} (1 - f_j(X_j)),$$

which concludes the proof of the theorem. ∎

Remark 2. Since for $f_N \equiv 1$, the stopping time $\tau_N = f_N(X_N)$ is optimal in $\mathcal{T}_N$, Theorem 1 inductively yields measurable functions $f_n \colon \mathbb{R}^d \to \{0,1\}$ such that for all $n \in \{0,1,\dots,N-1\}$, the stopping time τ_n given by (4) is optimal among $\mathcal{T}_n$. In particular,

$$\tau = \sum_{n=1}^{N} n f_n(X_n) \prod_{k=0}^{n-1} (1 - f_k(X_k)) \tag{6}$$

is an optimal stopping time for problem (1).

Remark 3. In many applications, the Markov process X starts from a deterministic initial value $x_0 \in \mathbb{R}^d$. Then the function f_0 enters the representation (6) only through the value $f_0(x_0) \in \{0,1\}$; that is, at time 0, only a constant and not a whole function has to be learned.

2.2 Neural network approximation

Our numerical method for problem (1) consists in iteratively approximating optimal stopping decisions $f_n \colon \mathbb{R}^d \to \{0,1\}$, $n = 0,1,\dots,N-1$, by a neural network $f^\theta \colon \mathbb{R}^d \to \{0,1\}$ with parameter $\theta \in \mathbb{R}^q$. We do this by starting with the terminal stopping decision $f_N \equiv 1$ and proceeding by backward induction.

More precisely, let $n \in \{0,1,\dots,N-1\}$, and assume parameter values $\theta_{n+1}, \theta_{n+2}, \dots, \theta_N \in \mathbb{R}^q$ have been found such that $f^{\theta_N} \equiv 1$ and the stopping time

$$\tau_{n+1} = \sum_{k=n+1}^{N} k f^{\theta_k}(X_k) \prod_{j=n+1}^{k-1} (1 - f^{\theta_j}(X_j))$$

produces an expected value $\mathbb{E}\, g(\tau_{n+1}, X_{\tau_{n+1}})$ close to the optimum V_{n+1}. Since f^θ takes values in $\{0,1\}$, it cannot be continuous in θ unless it is constant. As a result, it does not directly lend itself to a gradient-based optimization method. So, as an intermediate step, we introduce a multilayer feedforward neural network $F^\theta \colon \mathbb{R}^d \to (0,1)$ of the form

$$F^\theta = \psi \circ a_3^\theta \circ \varphi_{q_2} \circ a_2^\theta \circ \varphi_{q_1} \circ a_1^\theta,$$

where

- q_1 and q_2 are positive integers specifying the number of nodes in the two hidden layers,
- $a_1^\theta \colon \mathbb{R}^d \to \mathbb{R}^{q_1}$, $a_2^\theta \colon \mathbb{R}^{q_1} \to \mathbb{R}^{q_2}$ and $a_3^\theta \colon \mathbb{R}^{q_2} \to \mathbb{R}$ are affine functions,
- for $j \in \mathbb{N}$, $\varphi_j \colon \mathbb{R}^j \to \mathbb{R}^j$ is the component-wise ReLU activation function given by $\varphi_j(x_1,\dots,x_j) = (x_1^+,\dots,x_j^+)$
- $\psi \colon \mathbb{R} \to (0,1)$ is the standard logistic function given by $\psi(x) = e^x/(1+e^x) = 1/(1+e^{-x})$.

The components of the parameter $\theta \in \mathbb{R}^q$ of F^θ consist of the entries of the matrices $A_1 \in \mathbb{R}^{q_1 \times d}$, $A_2 \in \mathbb{R}^{q_2 \times q_1}$, $A_3 \in \mathbb{R}^{1 \times q_2}$ and the vectors $b_1 \in \mathbb{R}^{q_1}$, $b_2 \in \mathbb{R}^{q_2}$, $b_3 \in \mathbb{R}$ given by the representation of the affine functions

$$a_i^\theta(x) = A_i x + b_i, \quad i = 1, 2, 3.$$

So the dimension of the parameter space is $q = q_1(d + q_2 + 1) + 2q_2 + 1$, and for given $x \in \mathbb{R}^d$, $F^\theta(x)$ is almost everywhere smooth in θ. Our aim is to determine $\theta_n \in \mathbb{R}^q$ so that

$$\mathbb{E}\Big[g(n, X_n)F^{\theta_n}(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - F^{\theta_n}(X_n))\Big]$$

is close to the supremum $\sup_{\theta \in \mathbb{R}^q} \mathbb{E}\big[g(n, X_n)F^\theta(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - F^\theta(X_n))\big]$. Once this has been achieved, we define the function $f^{\theta_n} \colon \mathbb{R}^d \to \{0, 1\}$ by

$$f^{\theta_n} = I_{[0,\infty)} \circ a_3^{\theta_n} \circ \varphi_{q_2} \circ a_2^{\theta_n} \circ \varphi_{q_1} \circ a_1^{\theta_n}, \tag{7}$$

where $I_{[0,\infty)} \colon \mathbb{R} \to \{0, 1\}$ is the indicator function of $[0, \infty)$. The only difference between F^{θ_n} and f^{θ_n} is the final nonlinearity. While F^{θ_n} produces a stopping probability in $(0, 1)$, the output of f^{θ_n} is a true stopping decision given by 0 or 1, depending on whether F^{θ_n} takes a value below or above $1/2$.

The following result shows that for sufficiently large numbers q_1 and q_2 of hidden nodes, the neural network f^θ is flexible enough to make close to optimal stopping decisions.

Proposition 4. *Let $n \in \{0, 1, \dots, N-1\}$ and fix a stopping time $\tau_{n+1} \in \mathcal{T}_{n+1}$. Then, for every constant $\varepsilon > 0$, there exist positive integers q_1 and q_2 such that*

$$\begin{aligned}&\sup_{\theta \in \mathbb{R}^q} \mathbb{E}\Big[g(n, X_n)f^\theta(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - f^\theta(X_n))\Big] \\ &\geq \sup_{f \in \mathcal{D}} \mathbb{E}\big[g(n, X_n)f(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - f(X_n))\big] - \varepsilon,\end{aligned}$$

where $\mathcal{D}$ is the set of all measurable functions $f \colon \mathbb{R}^d \to \{0, 1\}$.

Proof. Fix $\varepsilon > 0$. It follows from the integrability condition (2) that there exists a measurable function $\tilde{f} \colon \mathbb{R}^d \to \{0, 1\}$ such that

$$\begin{aligned}&\mathbb{E}\Big[g(n, X_n)\tilde{f}(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - \tilde{f}(X_n))\Big] \\ &\geq \sup_{f \in \mathcal{D}} \mathbb{E}\big[g(n, X_n)f(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - f(X_n))\big] - \varepsilon/4.\end{aligned} \tag{8}$$

$\tilde{f}$ can be written as $\tilde{f} = I_A$ for the Borel set $A = \{x \in \mathbb{R}^d : \tilde{f}(x) = 1\}$. Moreover, by (2),

$$B \mapsto \mathbb{E}[|g(n, X_n)|I_B(X_n)] \quad \text{and} \quad B \mapsto \mathbb{E}\big[|g(\tau_{n+1}, X_{\tau_{n+1}})|I_B(X_n)\big]$$

define finite Borel measures on $\mathbb{R}^d$. Since every finite Borel measure on $\mathbb{R}^d$ is tight (see e.g. [1]), there exists a compact subset $K \subseteq A$ such that

$$\begin{aligned} &\mathbb{E}\big[g(n, X_n) I_K(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - I_K(X_n))\big] \\ &\geq \mathbb{E}\Big[g(n, X_n)\tilde{f}(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - \tilde{f}(X_n))\Big] - \varepsilon/4. \end{aligned} \tag{9}$$

Let $\rho_K \colon \mathbb{R}^d \to [0, \infty)$ be the distance function given by $\rho_K(x) = \inf_{y \in K} \|x - y\|_2$. Then

$$k_j(x) = \max\{1 - j\rho_K(x), -1\}, \quad j \in \mathbb{N},$$

defines a sequence of continuous functions $k_j \colon \mathbb{R}^d \to [-1, 1]$ that converge pointwise to $I_K - I_{K^c}$. So it follows from Lebesgue's dominated convergence theorem that there exists a $j \in \mathbb{N}$ such that

$$\begin{aligned} &\mathbb{E}\Big[g(n, X_n)\, I_{\{k_j(X_n) \geq 0\}} + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - I_{\{k_j(X_n) \geq 0\}})\Big] \\ &\geq \mathbb{E}\big[g(n, X_n) I_K(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - I_K(X_n))\big] - \varepsilon/4. \end{aligned} \tag{10}$$

By Theorem 1 of [35], k_j can be approximated uniformly on compacts by functions of the form

$$\sum_{i=1}^{r} (v_i^T x + c_i)^+ - \sum_{i=1}^{s} (w_i^T x + d_i)^+ \tag{11}$$

for $r, s \in \mathbb{N}$, $v_1, \ldots, v_r, w_1, \ldots, w_s \in \mathbb{R}^d$ and $c_1, \ldots, c_r, d_1, \ldots, d_s \in \mathbb{R}$. So there exists a function $h \colon \mathbb{R}^d \to \mathbb{R}$ which can be written as in (11) such that

$$\begin{aligned} &\mathbb{E}\big[g(n, X_n)\, I_{\{h(X_n) \geq 0\}} + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - I_{\{h(X_n) \geq 0\}})\big] \\ &\geq \mathbb{E}\Big[g(n, X_n)\, I_{\{k_j(X_n) \geq 0\}} + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - I_{\{k_j(X_n) \geq 0\}})\Big] - \varepsilon/4. \end{aligned} \tag{12}$$

Clearly, the composite mapping $I_{[0,\infty)} \circ h$ can be expressed as a neural net f^θ of the form (7) for suitable integers q_1, q_2 and a parameter value $\theta \in \mathbb{R}^q$. Hence, one obtains from (8), (9), (10) and (12) that

$$\begin{aligned} &\mathbb{E}\Big[g(n, X_n)\, f^\theta(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - f^\theta(X_n))\Big] \\ &\geq \sup_{f \in \mathcal{D}} \mathbb{E}\big[g(n, X_n) f(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - f(X_n))\big] - \varepsilon, \end{aligned}$$

and the proof is complete. □

The following is an immediate consequence of Theorem 1 and Proposition 4:

Corollary 5. *For a given optimal stopping problem of the form* (1) *and a constant* $\varepsilon > 0$*, there exist positive integers* q_1, q_2 *and functions* $f^{\theta_0}, f^{\theta_1}, \ldots, f^{\theta_N} : \mathbb{R}^d \to \{0,1\}$ *of the form* (7) *such that* $f^{\theta_N} \equiv 1$ *and the stopping time*

$$\hat{\tau} = \sum_{n=1}^{N} n f^{\theta_n}(X_n) \prod_{k=0}^{n-1} (1 - f^{\theta_k}(X_k))$$

satisfies $\mathbb{E}\, g(\hat{\tau}, X_{\hat{\tau}}) \geq \sup_{\tau \in \mathcal{T}} \mathbb{E}\, g(\tau, X_\tau) - \varepsilon$.

2.3 Parameter optimization and testing of the trained stopping rule

We train neural networks of the form (7) with given numbers q_1 and q_2 of nodes in the two hidden layers[2]. To numerically find parameters $\theta_n \in \mathbb{R}^q$ yielding good stopping decisions f^{θ_n} for all times $n \in \{0, 1, \ldots, N-1\}$, we approximate expected values with averages of Monte Carlo samples calculated from simulated paths of the process $(X_n)_{n=0}^N$.

Let $(x_n^m)_{n=0}^N$, $m = 1, 2, \ldots$ be independent realizations of such paths. We choose $\theta_N \in \mathbb{R}^q$ such that $f^{\theta_N} \equiv 1$ and determine $\theta_n \in \mathbb{R}^q$ for $n \leq N-1$ recursively. So suppose that for a given $n \in \{0, 1, \ldots, N-1\}$, parameters $\theta_{n+1}, \ldots, \theta_N \in \mathbb{R}^q$, have been found so that the stopping decisions $f^{\theta_{n+1}}, \ldots, f^{\theta_N}$ generate a stopping time

$$\tau_{n+1} = \sum_{k=n+1}^{N} k f^{\theta_k}(X_k) \prod_{j=n+1}^{k-1} (1 - f^{\theta_j}(X_j))$$

with corresponding expectation $\mathbb{E}\, g(\tau_{n+1}, X_{\tau_{n+1}})$ close to the optimal value V_{n+1}. If $n = N-1$, then $\tau_{n+1} = N$, and if $n \leq N-2$, τ_{n+1} can be written as

$$\tau_{n+1} = l_{n+1}(X_{n+1}, \ldots, X_{N-1})$$

for a measurable function $l_{n+1} \colon \mathbb{R}^{d(N-n-1)} \to \{n+1, n+2, \ldots, N\}$. Accordingly, denote

$$l_{n+1}^m = \begin{cases} N & \text{if } n = N-1 \\ l_{n+1}(x_{n+1}^m, \ldots, x_{N-1}^m) & \text{if } n \leq N-2 \end{cases}.$$

If at time n, one applies the continuous stopping decision F^θ and afterward behaves according to $f^{\theta_{n+1}}, \ldots, f^{\theta_N}$, the realized reward along the m-th simulated path of X is

$$r_n^m(\theta) = g(n, x_n^m) F^\theta(x_n^m) + g(l_{n+1}^m, x_{l_{n+1}^m}^m)(1 - F^\theta(x_n^m)).$$

[2]For a given application, one can try out different choices of q_1 and q_2 to find a suitable trade-off between accuracy and efficiency. Alternatively, the determination of q_1 and q_2 could be built into the training algorithm.

For large $M \in \mathbb{N}$,

$$\frac{1}{M}\sum_{m=1}^{M} r_n^m(\theta) \tag{13}$$

approximates the expected value

$$\mathbb{E}\Big[g(n, X_n)F^\theta(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - F^\theta(X_n))\Big].$$

Since $r_n^m(\theta)$ is almost everywhere smooth in θ, a stochastic gradient ascent method can be applied to find an approximate optimizer $\theta_n \in \mathbb{R}^q$ of (13). In the numerical examples in Section 3 below, we employed mini-batch gradient ascent with Xavier initialization [23], batch normalization [26] and Adam updating [28].

The same simulations $(x_n^m)_{n=0}^N$, $m = 1, 2, \ldots$ can be used to train the stopping decisions f^{θ_n} at all times $n \in \{0, 1, \ldots, N-1\}$. But to test the trained decision rules, a second set of independent realizations $(y_n^m)_{n=0}^N$, $m = 1, 2, \ldots, M$, of $(X_n)_{n=0}^N$ has to be simulated. The trained stopping time is of the form

$$\hat{\tau} = \sum_{n=1}^{N} n f^{\theta_n}(X_n) \prod_{k=0}^{n-1}(1 - f^{\theta_k}(X_k)). \tag{14}$$

So it can be expressed as $\hat{\tau} = l(X_0, \ldots, X_{N-1})$ for a measurable function $l\colon \mathbb{R}^{dN} \to \{0, 1, \ldots, N\}$. Denote $l^m = l(y_0^m, \ldots, y_{N-1}^m)$. We use the Monte Carlo approximation

$$\hat{V} = \frac{1}{M}\sum_{m=1}^{M} g(l^m, y_{l^m}^m)$$

to $\mathbb{E}\, g(\hat{\tau}, X_{\hat{\tau}})$ as our estimate of $\sup_{\tau\in\mathcal{T}} \mathbb{E}\, g(\tau, X_\tau)$.

For $\mathbb{E}\, g(\hat{\tau}, X_{\hat{\tau}})$, $\hat{V}$ is an unbiased estimate whose accuracy can be assessed with standard methods. By the law of large numbers, $\hat{V}$ converges to $\mathbb{E}\, g(\hat{\tau}, X_{\hat{\tau}})$ for $M \to \infty$. Moreover, assuming $g(\hat{\tau}, X_{\hat{\tau}})$ has finite variance[3], it can be derived from the central limit theorem that for any given $\alpha \in (0, 1)$,

$$\left[\hat{V} - z_{\alpha/2}\frac{\hat{\sigma}}{\sqrt{M}}, \hat{V} + z_{\alpha/2}\frac{\hat{\sigma}}{\sqrt{M}}\right] \tag{15}$$

is an asymptotically valid $1-\alpha$ confidence interval for $\mathbb{E}\, g(\hat{\tau}, X_{\hat{\tau}})$, where $z_{\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution and $\hat{\sigma}$ is the sample standard deviation given by

$$\hat{\sigma} = \sqrt{\frac{1}{M-1}\sum_{m=1}^{M}\Big(g(l^m, y_{l^m}^m) - \hat{V}\Big)^2}$$

[3]This follows for instance, if $\mathbb{E}\big[g(n, X_n)^2\big] < \infty$ for all $n \in \{0, 1, \ldots, N\}$.

(see, e.g., [21, Appendix A.2]).

On the other hand, as an estimate of $\sup_{\tau\in\mathcal{T}} \mathbb{E}\, g(\tau, X_\tau)$, $\hat{V}$ is a low-biased, and its accuracy depends on the quality of the trained stopping time $\hat{\tau}$. However, in the examples of Section 3 below, it yields very good results.

Remark 6. If the Markov process X starts from a deterministic initial value $x_0 \in \mathbb{R}^d$, the initial stopping decision is given by a constant $I_0 \in \{0, 1\}$. To learn I_0 from simulated paths of X, it is enough to compare the initial reward $g(0, x_0)$ to a Monte Carlo estimate $\hat{V}_1$ of $\mathbb{E}\, g(\tau_1, X_{\tau_1})$, where $\tau_1 \in \mathcal{T}_1$ is of the form

$$\tau_1 = \sum_{n=1}^{N} n f^{\theta_n}(X_n) \prod_{k=1}^{n-1} (1 - f^{\theta_k}(X_k))$$

for $f^{\theta_N} \equiv 1$ and trained parameters $\theta_1, \dots, \theta_{N-1} \in \mathbb{R}^q$. Then one sets $I_0 = 1$ (that is, stop immediately) if $g(0, x_0) \geq \hat{V}_1$ and $I_0 = 0$ (continue) otherwise. The resulting stopping time is of the form

$$\hat{\tau} = \begin{cases} 0 & \text{if } I_0 = 1 \\ \tau_1 & \text{if } I_0 = 0 \end{cases}.$$

3 Examples

In this section we test[4] our method on two examples: the pricing of a Bermudan max-call option on different underlying assets and the problem of optimally stopping a fractional Brownian motion.

3.1 Bermudan max-call options

Early exercise max-call options are one of the most studied examples in the numerics literature for optimal stopping problems; see, e.g., [36, 39, 20, 11, 24, 12, 3, 10, 9, 5, 6, 25, 34]. Their payoff depends on the maximum of d underlying assets.

Assume the assets evolve according to a multi-dimensional Black–Scholes model

$$S_t^i = s_0^i \exp\big([r - \delta_i - \sigma_i^2/2]t + \sigma_i W_t^i\big), \quad i = 1, 2, \dots, d, \tag{16}$$

for initial values $s_0^i \in (0, \infty)$, a risk-free interest rate $r \in \mathbb{R}$, dividend yields $\delta_i \in [0, \infty)$, volatilities $\sigma_i \in (0, \infty)$ and a d-dimensional Brownian motion W with constant instantaneous correlations[5] $\rho_{ij} \in \mathbb{R}$ between different components W^i and W^j. A Bermudan max-call option on $S^1, S^2, \dots, S^d$

[4] All computations were performed in single precision (float32) on a NVIDIA GeForce GTX 1080 GPU with 1974 MHz core clock and 8 GB GDDR5X memory with 1809.5 MHz clock rate. The underlying system consisted of an AMD Ryzen 1700X 3.4 GHz CPU with 32 GB DDR4-2400 memory running Tensorflow 1.5 on Ubuntu 16.04.

[5] That is, $\mathbb{E}[(W_t^i - W_s^i)(W_t^j - W_s^i)] = \rho_{ij}(t - s)$ for all $i \neq j$ and $s < t$.

has payoff $\left(\max_{1\le i\le d} S_t^i - K\right)^+$ and can be exercised at any point of a time grid $0 = t_0 < t_1 < \cdots < t_N$. Its price is given by

$$\sup_\tau \mathbb{E}\left[e^{-r\tau}\left(\max_{1\le i\le d} S_\tau^i - K\right)^+\right],$$

where the supremum is over all S-stopping times taking values in $\{t_0, t_1, \ldots, t_N\}$ (see, e.g., [42]). Denote $X_n^i = S_{t_n}^i$, $n = 1, 2, \ldots, N$, and let $\mathcal{T}$ be the set of X-stopping times. Then the price can be written as $\sup_{\tau\in\mathcal{T}} \mathbb{E}\, g(\tau, X_\tau)$ for

$$g(n, x) = e^{-rt_n}\left(\max_{1\le i\le d} x^i - K\right)^+,$$

and it is straight-forward to simulate $(X_n)_{n=0}^N$.

In the following we assume the time grid to be of the form $t_n = nT/N$, $n = 0, 1, \ldots, N$, for a maturity $T > 0$ and $N+1$ equidistant possible exercise dates. Even though $g(n, X_n)$ does not carry any information that is not already contained in X_n, our method worked more efficiently when we trained the optimal stopping decisions on Monte Carlo simulations of the $d+1$-dimensional Markov process $(Y_n)_{n=0}^N = (X_n, g(n, X_n))_{n=0}^N$ instead of $(X_n)_{n=0}^N$. So we first trained stopping times $\tau_1 \in \mathcal{T}_1$ of the form

$$\tau_1 = \sum_{n=1}^N n f^{\theta_n}(Y_n) \prod_{k=1}^{n-1}(1 - f^{\theta_k}(Y_k))$$

for $f^{\theta_N} \equiv 1$ and $f^{\theta_1}, \ldots, f^{\theta_{N-1}} \colon \mathbb{R}^{d+1} \to \{0, 1\}$ given by (7) with $d+1$ in place of d and $q_1 = q_2 = d + 50$. Then we determined our candidate optimal stopping times as

$$\hat\tau = \begin{cases} 0 & \text{if } I_0 = 1 \\ \tau_1 & \text{if } I_0 = 0 \end{cases}$$

for a constant $I_0 \in \{0, 1\}$ depending[6] on whether it was optimal to stop immediately at time 0 or not (see Remark 6 above).

Symmetric case

First, we considered the special case, where $s_0^i = s_0$, $\delta_i = \delta$, $\sigma_i = \sigma$ for every $i \in \{1, 2, \ldots, d\}$ and $\rho_{ij} = \rho$ for all $i \neq j$. The results are reported in Table 1.

Asymmetric case

As a second example we considered model (16) with $s_0^i = s_0$, $\delta_i = \delta$ for every $i = \{1, 2, \ldots, d\}$ and $\rho_{ij} = \rho$ for all $i \neq j$, but different volatilities $\sigma_1 < \sigma_2 < \cdots < \sigma_d$. For $d = 5$, we chose the same specification as [10]: $\sigma_i = i \times 0.08$, $i = 1, 2, \ldots, 5$. For $d > 5$, we set $\sigma_i = 0.1 + i \times 0.5/d$, $i = 1, 2, \ldots, d$. Our results are reported in Table 2.

[6] In fact, in none of the examples in this paper it is optimal to stop at time 0. So $\hat\tau = \tau_1$ in all these cases.

d	s_0	Point Est.	95% CI	Comp. Time	BC 95% CI
5	90	16.635	[16.619, 16.650]	42.3	[16.620, 16.653]
5	100	26.148	[26.129, 26.167]	42.1	[26.115, 26.164]
5	110	36.770	[36.748, 36.790]	42.0	[36.710, 36.798]
10	90	26.195	[26.176, 26.213]	44.7	
10	100	38.313	[38.292, 38.334]	44.7	
10	110	50.845	[50.822, 50.868]	44.7	
20	90	37.685	[37.665, 37.704]	50.2	
20	100	51.567	[51.545, 51.589]	50.1	
20	110	65.526	[65.502, 65.550]	50.4	
30	90	44.819	[44.798, 44.839]	56.3	
30	100	59.506	[59.484, 59.528]	56.4	
30	110	74.213	[74.189, 74.237]	56.2	
50	90	53.883	[53.863, 53.904]	70.4	
50	100	69.551	[69.528, 69.574]	70.6	
50	110	85.255	[85.230, 85.279]	70.4	
100	90	66.353	[66.332, 66.374]	108.9	
100	100	83.405	[83.382, 83.428]	108.9	
100	110	100.421	[100.396, 100.447]	108.9	
200	90	78.987	[78.966, 79.009]	188.1	
200	100	97.390	[97.366, 97.413]	188.2	
200	110	115.830	[115.804, 115.856]	188.3	
500	90	95.965	[95.943, 95.987]	533.5	
500	100	116.256	[116.232, 116.281]	533.1	
500	110	136.522	[136.495, 136.549]	534.1	

Table 1: Summary results for max-call options on d symmetric assets for parameter values of $r = 5\%$, $\delta = 10\%$, $\sigma = 20\%$, $\rho = 0$, $K = 100$, $T = 3$, $N = 9$. 95% CI is the 95% confidence interval (15) for $\mathbb{E}\, g(\hat{\tau}, X_{\hat{\tau}})$, where $\hat{\tau}$ is our candidate optimal stopping time. Computing times are reported in seconds. BC 95% CI is the 95% confidence interval for $\sup_{\tau \in \mathcal{T}} \mathbb{E}\, g(\tau, X_\tau)$ computed in [10].

d	s_0	Point Est.	95% CI	Comp. Time	BC 95% CI
5	90	27.659	[27.627, 27.690]	42.1	[27.468, 27.686]
5	100	37.976	[37.940, 38.012]	42.1	[37.730, 38.020]
5	110	49.488	[49.448, 49.528]	42.0	[49.155, 49.531]
10	90	85.913	[85.833, 85.993]	44.8	
10	100	104.630	[104.541, 104.719]	44.7	
10	110	123.727	[123.629, 123.825]	44.7	
20	90	126.00	[125.903, 126.097]	50.6	
20	100	149.621	[149.513, 149.730]	50.7	
20	110	173.340	[173.221, 173.459]	50.4	
30	90	154.475	[154.366, 154.583]	56.6	
30	100	181.317	[181.196, 181.437]	56.4	
30	110	208.157	[208.024, 208.290]	56.4	
50	90	196.059	[195.934, 196.183]	70.9	
50	100	227.483	[227.345, 227.622]	70.7	
50	110	258.720	[258.568, 258.872]	70.7	
100	90	263.293	[263.143, 263.443]	108.8	
100	100	302.084	[301.918, 302.251]	108.7	
100	110	340.766	[340.584, 340.949]	108.7	
200	90	344.503	[344.323, 344.682]	189.0	
200	100	392.109	[391.911, 392.307]	188.2	
200	110	440.058	[439.839, 440.276]	188.8	
500	90	476.346	[476.121, 476.571]	534.4	
500	100	538.657	[538.407, 538.907]	533.6	
500	110	601.419	[601.146, 601.693]	534.5	

Table 2: Summary results for max-call options on d asymmetric assets for parameter values of $r = 5\%$, $\delta = 10\%$, $\rho = 0$, $K = 100$, $T = 3$, $N = 9$. 95% CI is the 95% confidence interval (15) for $\mathbb{E}\, g(\hat{\tau}, X_{\hat{\tau}})$, where $\hat{\tau}$ is our candidate optimal stopping time. Computing times are in seconds. BC 95 % CI is the 95% confidence interval for $\sup_{\tau \in \mathcal{T}} \mathbb{E}\, g(\tau, X_\tau)$ computed in [10].

3.2 Optimally stopping a fractional Brownian motion

A fractional Brownian motion with Hurst parameter $H \in (0,1]$ is a continuous centered Gaussian process $(W_t^H)_{t\geq 0}$ with covariance structure

$$\mathbb{E}[W_t^H W_s^H] = \frac{1}{2}\left(t^{2H} + s^{2H} - |t-s|^{2H}\right)$$

(see, e.g., [37, 41]). For $H = 1/2$, W^H is a standard Brownian motion. So, by the optional stopping theorem, $\mathbb{E}\, W_\tau^{1/2} = 0$ for every $W^{1/2}$-stopping time τ bounded above by a constant (see, e.g., [22]). However, for $H \neq 1/2$, the increments of W^H are correlated – positively correlated for $H \in (1/2, 1]$ and negatively correlated for $H \in (0, 1/2)$. In both cases, W^H is neither a martingale nor a Markov process, and there exist bounded W^H-stopping times τ such that $\mathbb{E}\, W_\tau^H > 0$; see, e.g., [32] for two classes of simple stopping rules $0 \leq \tau \leq 1$ and simulations of the corresponding expected values $\mathbb{E}\, W_\tau^H$.

In order to approximate the supremum

$$\sup_{0\leq\tau\leq 1} \mathbb{E}\, W_\tau^H \tag{17}$$

over all W^H-stopping times $0 \leq \tau \leq 1$, we denote $t_n = n/100$, $n = 0, 1, 2, \ldots, 100$, and introduce the 100-dimensional Markov process $(X_n)_{n=0}^{100}$ given by

$$\begin{aligned} X_0 &= (0, 0, \ldots, 0) \\ X_1 &= (W_{t_1}^H, 0, \ldots, 0) \\ X_2 &= (W_{t_2}^H, W_{t_1}^H, 0, \ldots, 0) \\ &\vdots \\ X_{100} &= (W_{t_{100}}^H, W_{t_{99}}^H, \ldots, W_{t_1}^H). \end{aligned}$$

The discretized stopping problem

$$\sup_{\tau\in\mathcal{T}} \mathbb{E}\, g(X_\tau), \tag{18}$$

where $\mathcal{T}$ is the set of all X-stopping times and $g\colon \mathbb{R}^{100} \to \mathbb{R}$ the function given by $g(x^1, \ldots, x^{100}) = x^1$, approximates (17) from below.

For our numerical approximation of (18), we trained networks of the form (7) with $d = 100$, $q_1 = 110$, and $q_2 = 55$. To simulate paths of $(X_n)_{n=0}^{100}$, we used the fact that the increments $Y_n^H = W_{t_n}^H - W_{t_{n-1}}^H$, $n = 1, 2, \ldots, N$, form a centered stationary Gaussian sequence (so-called fractional Gaussian noise) with autocovariance

$$\mathbb{E}[Y_n^H Y_{n+k}^H] = \frac{|k+1|^{2H} - |k|^{2H} + |k-1|^{2H}}{2 \times 100^{2H}}.$$

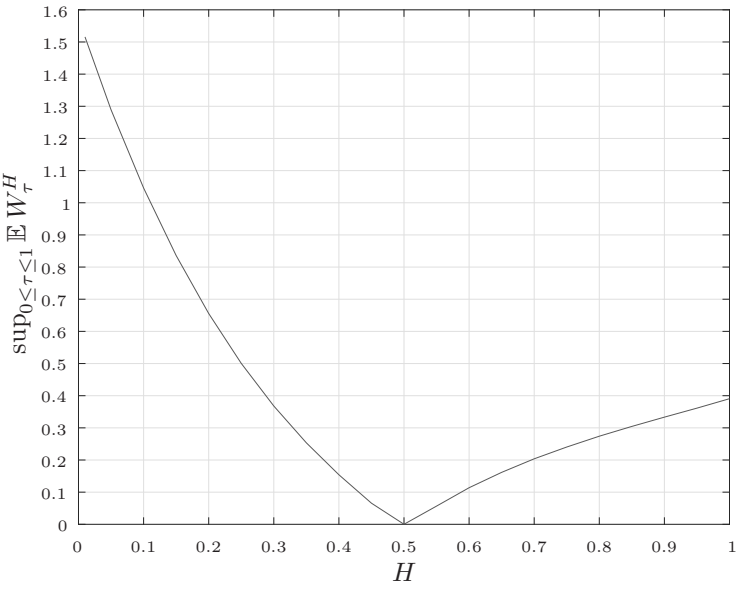

Figure 1: Estimates of $\sup_{0\le\tau\le1}\mathbb{E}\,W_\tau^H$ for different values of H. For each choice of H, the computation took about 500 seconds. The 95% confidence intervals were of the order of 10^{-3}.

So $(Y_1^H,\dots,Y_{100}^H)$ can efficiently be simulated by embedding its covariance matrix in a positive definite symmetric circulant matrix[7], which can be diagonalized using fast Fourier transformation (see [14, 45, 18]). From there, sample paths of $(X_n)_{n=0}^{100}$ are obtained via $W_{t_n}^H=\sum_{k=1}^n Y_k^H$. We computed estimates of (18) for $H=0.01,0.05,0.1,0.15,\dots,1$. The results are shown in Figure 1. It can be seen that for $H>1/2$, our methods yields up to three times higher expected payoffs than the heuristic stopping rules of [32]. For $H<1/2$, they are up to five times higher.

References

[1] C.D. Aliprantis and K.C. Border (2006). Infinite Dimensional Analysis. 3rd Edition. Springer.

[2] L. Andersen (2000). A simple approach to the pricing of Bermudan swaptions in the multi-factor LIBOR market model. Journal of Computational Finance 3(2), 5–32.

[7] The covariance matrix of $(Y_1^H,\dots,Y_{100}^H)$ is a positive definite symmetric Toeplitz matrix, and it is shown in [16] that each matrix of this form can be embedded in a sufficiently large positive definite symmetric circulant matrix. Moreover, numerical results show that for all $H\in\{0.01,0.05,0.1,0.15,\dots,1\}$, the minimal circulant embedding of the covariance matrix of $(Y_1^H,\dots,Y_{100}^H)$ is already positive definite; see also [43, 31].

[3] L. Andersen and M. Broadie (2004). Primal-dual simulation algorithm for pricing multidimensional American options. Management Science 50(9), 1222–1234.

[4] V. Bally, G. Pagés and J. Printems (2005). A quantization tree method for pricing and hedging multidimensional American options. Math. Fin. 15(1), 119–168.

[5] D. Belomestny (2011). On the rates of convergence of simulation-based optimization algorithms for optimal stopping problems. Ann. Appl. Prob.

[6] D. Belomestny (2013). Solving optimal stopping problems via empirical dual optimization. Ann. App. Probability

[7] D. Belomestny, C. Bender and J. Schoenmakers (2009). True upper bounds for Bermudan products via non-nested Monte Carlo. Math. Fin. 19(1), 53–71.

[8] D Belomestny, J Schoenmakers and F Dickmann (2013). Multilevel dual approach for pricing American style derivatives. Fin. Stoch.

[9] S.J. Berridge and J.M. Schumacher (2008). An irregular grid approach for pricing high-dimensional American options. Journal of Computational and Applied Mathematics 222(1), 94–111.

[10] M. Broadie and M. Cao (2008). Improved lower and upper bound algorithms for pricing American options by simulation. Quantitative Finance, 8, 845–861.

[11] P.P. Boyle, A.W. Kolkiewicz and K.S. Tan (2003). An improved simulation method for pricing high-dimensional American derivatives. Mathematics and Computers in Simulation 62, 315–322.

[12] M. Broadie and P. Glasserman (2004). A stochastic mesh method for pricing high-dimensional American options. Journal of Computational Finance, 7(4), 35–72.

[13] N. Chen and P. Glasserman (2007). Additive and multiplicative duals for American option pricing. Finance Stoch., 11, 153–179.

[14] R. Davis and D.S. Harte (1987). Tests for Hurst effect. Biometrika 74, 95–101.

[15] M. Davis and I. Karatzas (1994). A deterministic approach to optimal stopping. In F. P. Kelly, editor, Probability, Statistics and Optimization: A Tribute to Peter Whittle, pages 455–466. J. Wiley and Sons.

[16] A. Dembo, C.L. Mallows and L.A. Shepp (1989). Embedding nonnegative definite Toeplitz matrices in nonnegative definite circulant matrices, with application to covariance estimation. IEEE Trans. Inform. Theory, IT-35, 1206–1212.

[17] V.V. Desai, V.F. Farias and C.C. Moallemi (2012). Pathwise optimization for optimal stopping problems.

[18] C.R. Dietrich and G.N. Newsam (1997). Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix. SIAM Journal on Scientific Computing 18(4), 1088–1107.

[19] D. Egloff, M. Kohler and N. Todorovic (2007). A dynamic look-ahead Monte Carlo algorithm for pricing Bermudan options. Ann. Appl. Probab. 17(4), 1138–1171.

[20] D. García (2003). Convergence and biases of Monte Carlo estimates of American option prices using a parametric exercise rule. Journal of Economic Dynamics and Control 27, 1855–1879.

[21] P. Glasserman (2013). Monte Carlo Methods in Financial Engineering. Springer Science and Media.

[22] G. Grimmett and D. Stirzaker (2001). Probability and Random Processes. Third Edition. Oxford University Press.

[23] X. Glorot and Y. Bengio (2010). Understanding the difficulty of training deep feedforward neural networks. PMLR 9, 249–256.

[24] M. Haugh and L. Kogan (2004). Pricing American options: a duality approach. Operations Research, 52, 258–270.

[25] S. Jain and C.W. Oosterlee (2015). The Stochastic Grid Bundling Method: Efficient pricing of Bermudan options and their Greeks. Applied Mathematics and Computation 269, 412–431.

[26] S. Ioffe and C. Szegedy (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. Proceedings of The 32nd International Conference on Machine Learning.

[27] F. Jamshidian (2007). The duality of optimal exercise and domineering claims: a Doob–Meyer decomposition approach to the snell envelope. Stochastics An International Journal of Probability and Stochastic Processes, 79(1-2), 27–60.

[28] D. Kingma and J. Ba (2015). Adam: A method for stochastic optimization, ICLR.

[29] M. Kohler, A. Krzyzak and N. Todorovic (2010). Pricing of high-dimensional American options by neural networks. Math. Fin. 20(3), 383–410.

[30] A. Kolodko and J. Schoenmakers (2006). Iterative construction of the optimal Bermudan stopping time. Finance and Stoch. 10(1), 27–49.

[31] D.P. Kroese and Z.I. Botev (2015). Spatial process simulation. Stochastic Geometry, Spatial Statistics and Random Fields, 369-404. Springer.

[32] A.V. Kulikov and P.P. Gusyatnikov (2016). Stopping times for fractional Brownian motion. Computational Management Science. Lecture Notes in Economics and Mathematical Systems 682.

[33] D. Lamberton and B. Lapeyre (2000). Introduction to Stochastic Calculus Applied to Finance, Second Edition. Chapman and Hall/CRC, Boca Raton, Florida.

[34] J. Lelong (2016). Pricing American options using martingale bases. Arxiv Preprint.

[35] M. Leshno, V.Y. Lin, A. Pinkus and S. Schocken (1993). Multilayer feedforward networks with a non-polynomial activation function can approximate any function. Neural Networks 6, 861–867.

[36] F. Longstaff and E.S. Schwartz (2001). Valuing American options by simulation: a simple least-square approach. Review of Financial Studies, 14(1), 113–147.

[37] B. Mandelbrot and J.W. van Ness (1968). Fractional Brownian motions, fractional noises and applications. SIAM Review 10 (4), 422–437.

[38] G. Peskir and A. Shiryaev (2006). Optimal Stopping and Free-Boundary Problems. Lectures in Mathematics. ETH Zurich.

[39] L.C.G. Rogers (2002). Monte Carlo valuation of American options. Mathematical Finance 12(3), 271–286.

[40] L.C.G. Rogers (2010). Dual valuation and hedging of Bermudan options. SIAM Journal on Financial Mathematics, 1, 604–608.

[41] G. Samorodnitsky and M.S. Taqqu (1994). Stable Non-Gaussian Random Processes. Chapman & Hall.

[42] M. Schweizer (2002). On Bermudan options. Advances in Finance and Stochastics. Springer.

[43] G. Shevchenko (2015). Fractional Brownian motion in a nutshell. International Journal of Modern Physics: Conference Series 36.

[44] J. N. Tsitsiklis and B Van Roy (2001). Regression methods for pricing complex American-style options. IEEE Transactions on Neural Networks, 12(4), 694–703.

[45] A.T.A. Wood and G. Chan (1994). Simulation of stationary gaussian processes in $[0,1]^d$. Journal of Computational and Graphical Statistics 3(4), 409–432.