Neural network regression for Bermudan option pricing

Bernard Lapeyre, Jérôme Lelong

Neural network regression for Bermudan option pricing

Bernard Lapeyre* Jérôme Lelong †

November 27, 2020

Abstract

The pricing of Bermudan options amounts to solving a dynamic programming principle, in which the main difficulty, especially in high dimension, comes from the conditional expectation involved in the computation of the continuation value. These conditional expectations are classically computed by regression techniques on a finite dimensional vector space. In this work, we study neural networks approximations of conditional expectations. We prove the convergence of the well-known Longstaff and Schwartz algorithm when the standard least-square regression is replaced by a neural network approximation, assuming an efficient algorithm to compute this approximation. We illustrate the numerical efficiency of neural networks as an alternative to standard regression methods for approximating conditional expectations on several numerical examples.
Key words: Bermudan options, optimal stopping, regression methods, deep learning, neural networks.

1 Introduction

Solving the backward recursion involved in the computation american option prices has been a challenging problem for years and various approaches have been proposed to approximate its solution. The real difficulty lies in the computation of the conditional expectation $\mathbb{E}[U_{T_{n+1}}|\mathcal{F}_{T_n}]$ at each time step of the recursion. If we were to classify the different approaches, we could say that there are regression based approaches (see Tilley [1993], Carriere [1996], Tsitsiklis and Roy [2001], Broadie and Glasserman [2004]) and quantization approaches (see Bally and Pages [2003], Bronstein et al. [2013]). We refer to Bouchard and Warin [2012] and Pagès [2018] for an in depth survey of the different techniques to price Bermudan options.

*Université Paris-Est, Cermics (ENPC), INRIA, F-77455 Marne-la-Vallée, France
email: bernard.lapeyre@enpc.fr
†Univ. Grenoble Alpes, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.
email: jerome.lelong@univ-grenoble-alpes.fr

Among all these available algorithms to compute american option prices using the dynamic programming principle, the one proposed by Longstaff and Schwartz [2001] has the favour of many practitioners. Their approach is based on iteratively selecting an optimal policy. Here, we propose and analyse a version of this algorithm which uses neural networks in order to compute an approximation of the conditional expectation and then to obtain an optimal exercising policy.

The use of neural network for the computation of American option prices is not new but we are aware of no work specifically devoted to its use for LS-style algorithms (LS for Longstaff and Schwartz [2001]). In Haugh and Kogan [2004], the authors used neural networks in numercial experiments to price American options through the dynamic programing equation on the value function. This led them to a Tsitsiklis and Roy [2001]-type algorithm which is different from LS-type algorithm, studied in this paper, which involve only the optimal stopping policy. Kohler et al. [2010] used neural networks to price American options but they also used the dynamic programing equation on the value function. Moreover they used new samples of the whole path of the underlying process X at each time step n to prove the convergence. In our approach, we use a neural network inspired modification of the original Longstaff-Schwartz algorithm and we draw a set of M samples with the distribution of $(X_{T_0}, X_{T_1}, \dots, X_{T_N})$ before starting and we use these very same samples at each time step. This saves a lot of computational time by avoiding a very costly resimulation at each time step, which very much improves the efficiency of our approach. Deep learning was also used in the context of optimal stopping by Becker et al. [2019a,b] to parametrize the optimal policy.

Now, we describe the framework of our study. We fix some finite time horizon $T > 0$ and a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{0\le t\le T}, \mathbb{P})$ modeling a financial market with $\mathcal{F}_0$ being the trivial $\sigma-$algebra. We assume that the short interest rate is modeled by an adapted process $(r_t)_{0\le t\le T}$ and that $\mathbb{P}$ is an associated risk neutral measure. We consider a Bermudan option with exercising dates $0 = t_0 \le T_1 < T_2 < \cdots < T_N = T$ and discounted payoff $\tilde{Z}_{T_n}$ if exercised at time T_n. For convenience, we add 0 and T to the exercising dates. This is definitely not a requirement of the method we propose here but it makes notation lighter and avoids to deal with the purely European part involved in the Bermudan option. We assume that the discrete time discounted payoff process $(Z_{T_n})_{0\le n\le N}$ is adapted to the filtration $(\mathcal{F}_{T_n})_{0\le n\le N}$ and that $\mathbb{E}[\max_{0\le n\le N} |Z_{T_n}|^2] < \infty$.

In a complete market, if $\mathbb{E}$ denote the expectation under the risk neutral probability, standard arbitrage pricing arguments allows to define the discounted value $(U_n)_{0\le n\le N}$ of the Bermudan option at times $(T_n)_{0\le n\le N}$ by

$$U_{T_n} = \sup_{\tau\in\mathcal{T}_{T_n,T}} \mathbb{E}[Z_\tau|\mathcal{F}_{T_n}]. \tag{1}$$

Using the Snell enveloppe theory, the sequence U can be proved to be given by the following dynamic programing equation

$$\begin{cases} U_{T_N} &= Z_{T_N} \\ U_{T_n} &= \max\left(Z_{T_n}, \mathbb{E}[U_{T_{n+1}}|\mathcal{F}_{T_n}]\right), \quad 0 \le n \le N-1. \end{cases} \tag{2}$$

This equation can be rewritten in term of optimal policy. Let τ_n be the smallest optimal policy after time T_n — the smallest stopping time reaching the supremum in (1) — then

$$\begin{cases} \tau_N = T_N \\ \tau_n = T_n \mathbf{1}_{\left\{Z_{T_n} \geq \mathbb{E}[Z_{\tau_{n+1}} | \mathcal{F}_{T_n}]\right\}} + \tau_{n+1} \mathbf{1}_{\left\{Z_{T_n} < \mathbb{E}[Z_{\tau_{n+1}} | \mathcal{F}_{T_n}]\right\}}. \end{cases} \tag{3}$$

All these methods based on the dynamic programming principle either as value iteration (2) or policy iteration (3) require a Markovian setting to be implemented such that the conditional expectation knowing the whole past can be replaced by the conditional expectation knowing only the value of a Markov process at the current time. We assume that the discounted payoff process writes $Z_{T_n} = \phi_n(X_{T_n})$, for any $0 \leq n \leq N$, where $(X_t)_{0 \leq t \leq T}$ is an adapted Markov process taking values in $\mathbb{R}^r$. Hence, the conditional expectation involved in (3) simplifies into $\mathbb{E}[Z_{\tau_{n+1}} | \mathcal{F}_{T_n}] = \mathbb{E}[Z_{\tau_{n+1}} | X_{T_n}]$ and can therefore be approximated by a standard least square method.

Note that this setting allows to consider most standard financial models. For local volatility models, the process X is typically defined as $X_t = (r_t, S_t)$, where S_t is the price of an asset and r_t the instantaneous interest rate (only $X_t = S_t$ when the interest rate is deterministic). In the case of stochastic volatility models, X also includes the volatility process σ, $X_t = (r_t, S_t, \sigma_t)$. Some path dependent options can also fit in this framework at the expense of increasing the size of the process X. For instance, in the case of an Asian option with payoff $(\frac{1}{T} A_T - S_T)_+$ with $A_t = \int_0^t S_u du$, one can define X as $X_t = (r_t, S_t, \sigma_t, A_t)$ and then the Asian option can be considered as a vanilla option on the two dimensional but non tradable assets (S, A).

Once the Markov process X is identified, the conditional expectations can be written

$$\mathbb{E}[Z_{\tau_{n+1}} | \mathcal{F}_{T_n}] = \mathbb{E}[Z_{\tau_{n+1}} | X_{T_n}] = \psi_n(X_{T_n}) \tag{4}$$

where ψ_n solves the following minimization problem

$$\inf_{\psi \in L^2(\mathcal{L}(X_{T_n}))} \mathbb{E}\left[\left|Z_{\tau_{n+1}} - \psi(X_{T_n})\right|^2\right]$$

with $L^2(\mathcal{L}(X_{T_n}))$ being the set of all measurable functions f such that $\mathbb{E}[f(X_{T_n})^2] < \infty$. The real challenge comes from properly approximating the space $L^2(\mathcal{L}(X_{T_n}))$ by a finite dimensional space: one typically uses polynomials or local bases (see Gobet et al. [2005], Bouchard and Warin [2012]) and in any case it always boils down to a linear regression. In this work, we use neural networks to approximate ψ_n in (4). The main difference between neural networks and the regression approaches commonly used comes from the non linearity of neural networks, which also makes their strength. Note that the set of neural networks with a fixed number of layers and neurons is obviously not a vector space and not even convex. Through neural networks, this paper investigates the effects of using non linear approximations of conditional expectations in the Longstaff Schwartz algorithm.

The paper is organized as follows. In Section 2, we start with some preliminaries on neural networks and recall the universal approximation theorem. Then, in Section 3, we describe our algorithm, whose convergence is studied in Section 4. Finally, we present some numerical results in Section 5.

2 Preliminaries on deep neural network

Deep Neural networks (DNN) aim at approximating (complex non linear) functions defined on finite-dimensional spaces, and in contrast with the usual additive approximation theory built via basis functions, like polynomials, they rely on composition of layers of simple functions. The relevance of neural networks comes from the universal approximation theorem and the Kolmogorov-Arnold representation theorem (see Arnold [2009], Kolmogorov [1956], Cybenko [1989], Hornik [1991], Pinkus [1999]), and this has shown to be successful in numerous practical applications.

We consider the feed forward neural network — also called multilayer perceptron — for the approximation of the continuation value at each time step. From a mathematical point view, we can model a DNN by a non linear function

$$x \in \mathcal{X} \subset \mathbb{R}^r \longmapsto \Phi(x; \theta) \in \mathbb{R}$$

where Φ typically writes as function compositions. Let $L \geq 2$ be an integer, we write

$$\Phi = A_L \circ \sigma_a \circ A_{L-1} \circ \cdots \circ \sigma_a \circ A_1 \tag{5}$$

where for $\ell = 1, \ldots, L$, $A_\ell : \mathbb{R}^{d_{\ell-1}} \to \mathbb{R}^{d_\ell}$ are affine functions

$$A_\ell(x) = \mathcal{W}_\ell x + \beta_\ell \quad \text{for } x \in \mathbb{R}^{d_{\ell-1}},$$

with $\mathcal{W}_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$, and $\beta_\ell \in \mathbb{R}^{d_\ell}$. In our setting, we have $d_1 = r$ and $d_L = 1$. The function σ_a is often called the *activation function* and is applied component wise. The number d_ℓ of rows of the matrix $\mathcal{W}_\ell$ is usually interpreted as the number of neurons of layer ℓ. For the sake of simpler notation, we embed all the parameters of the different layers in a unique high dimensional parameter θ and $\theta = (\mathcal{W}_\ell, \beta_\ell)_{\ell=1,\ldots,L} \in \mathbb{R}^{N_d}$ with $N_d = \sum_{\ell=1}^{L} d_\ell(1 + d_{\ell-1})$.

Let $L > 0$ be fixed in the following, we introduce the set $\mathcal{NN}_\infty$ of all DNN of the above form. Now, we need to restrict the maximum number of neurons per layer. Let $p \in \mathbb{N}$, $p > 1$, we denote by $\mathcal{NN}_p$ the set of neural networks with at most p neurons per hidden layer and $L-1$ layers and bounded parameters. More precisely, we pick an increasing sequence of positive real numbers $(\gamma_p)_p$ such that $\lim_{p\to\infty} \gamma_p = \infty$. We introduce the set

$$\Theta_p = \{\theta \in \mathbb{R}^r \times \mathbb{R}^{p\times r} \times \left(\mathbb{R}^p \times \mathbb{R}^{p\times p}\right)^{L-2} \times \mathbb{R} \times \mathbb{R}^p \,:\, |\theta| \leq \gamma_p\}. \tag{6}$$

Then, $\mathcal{NN}_p$ is defined by

$$\mathcal{NN}_p = \{\Phi(\cdot; \theta) : \mathbb{R}^r \to \mathbb{R} \,;\, \theta \in \Theta_p\}$$

and we have $\mathcal{NN}_\infty = \cup_{p\in\mathbb{N}} \mathcal{NN}_p$. An element of $\mathcal{NN}_p$ with be denoted by $\Phi_p(\cdot; \theta)$ with $\theta \in \Theta_p$. Note that the space $\mathcal{NN}_p$ is not a vector space, nor a convex set and therefore finding the element of $\mathcal{NN}_p$ that best approximates a given function cannot be simply interpreted as an orthogonal projection.

The use of DNN as function approximations is justified by the fundamental results of Hornik [1991] (see also Pinkus [1999] for related results).

Theorem 2.1 (Universal Approximation Theorem) *Assume that the function σ_a is non constant and bounded. Let μ denote a probability measure on $\mathbb{R}^r$, then for any $L \geq 2$, $\mathcal{NN}_\infty$ is dense in $L^2(\mathbb{R}^r, \mu)$.*

Theorem 2.2 (Universal Approximation Theorem) *Assume that the function σ_a is a non constant, bounded and continuous function, then, when $L = 2$, $\mathcal{NN}_\infty$ is dense into $C(\mathbb{R}^r)$ for the topology of the uniform convergence on compact sets.*

Remark 2.3 We can rephrase Theorem (2.1) in terms of approximating random variables. Let Y be a real valued random variable defined on $(\Omega, \mathcal{A})$ s.t. $\mathbb{E}[Y^2] < \infty$. Let X be an other random variable defined on $(\Omega, \mathcal{A})$ taking values in $\mathbb{R}^r$ and $\mathcal{G} \subset \mathcal{A}$ the smallest $\sigma-$algebra such that X is $\mathcal{G}$ measurable. Then, there exists a sequence $(\theta_p)_{p\geq 2} \in \prod_{p=2}^{\infty} \Theta_p$, such that $\lim_{p\to\infty} \mathbb{E}[|Y - \Phi_p(X; \theta_p)|^2] = 0$. Therefore, if for every $p \geq 2$, $\alpha_p \in \Theta_p$ solves

$$\inf_{\theta\in\Theta_p} \mathbb{E}[|\Phi_p(X; \theta) - Y|^2],$$

then the sequence $(\Phi_p(X; \alpha_p))_{p\geq 2}$ converges to $\mathbb{E}[Y|\mathcal{G}]$ in $L^2(\Omega)$ when $p \to \infty$. Note that as long as the activation function σ_a is bounded, $\Phi_p(X; \alpha_p) \in L^2(\Omega)$ for every $p \geq 2$.

3 The algorithm

3.1 Description of the algorithm

We recall the dynamic programming principle on the optimal policy

$$\begin{cases} \tau_N = T_N \\ \tau_n = T_n \mathbf{1}_{\left\{Z_{T_n} \geq \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right\}} + \tau_{n+1} \mathbf{1}_{\left\{Z_{T_n} < \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right\}}, \text{ for } 1 \leq n \leq N-1. \end{cases}$$

Then, the time-0 price of the Bermudan option writes

$$U_0 = \max(Z_0, \mathbb{E}[Z_{\tau_1}]).$$

In order to solve this dynamic programming equation we need to compute a conditional expectation at each time step. The idea proposed by Longstaff and Schwartz [2001] was to approximate these conditional expectations by a regression problem on a well chosen set of functions. In this work, we use a DNN to perform this approximation.

$$\begin{cases} \tau_N^p = T_N^p \\ \tau_n^p = T_n \mathbf{1}_{\left\{Z_{T_n} \geq \Phi_p(X_{T_n}; \theta_n^p)\right\}} + \tau_{n+1} \mathbf{1}_{\left\{Z_{T_n} < \Phi_p(X_{T_n}; \theta_n^p)\right\}}, \text{ for } 1 \leq n \leq N-1 \end{cases} \quad (7)$$

where θ_n^p solves the following optimization problem

$$\inf_{\theta\in\Theta_p} \mathbb{E}\left[\left|\Phi_p(X_{T_n}; \theta) - Z_{\tau_{n+1}^p}\right|^2\right]. \quad (8)$$

Since the conditional expectation operator is an orthogonal projection, we have

$$\mathbb{E}\left[\left|\Phi_p(X_{T_n};\theta) - Z_{\tau^p_{n+1}}\right|^2\right] = \mathbb{E}\left[\left|\Phi_p(X_{T_n};\theta) - \mathbb{E}\left[Z_{\tau^p_{n+1}}|\mathcal{F}_{T_n}\right]\right|^2\right] + \mathbb{E}\left[\left|Z_{\tau^p_{n+1}} - \mathbb{E}\left[Z_{\tau^p_{n+1}}|\mathcal{F}_{T_n}\right]\right|^2\right].$$

Therefore, any minimizer in (8) is also a solution to the following minimization problem

$$\inf_{\theta\in\Theta_p} \mathbb{E}\left[\left|\Phi_p(X_{T_n};\theta) - \mathbb{E}\left[Z_{\tau^p_{n+1}}|\mathcal{F}_{T_n}\right]\right|^2\right]. \tag{9}$$

The standard approach is to sample a bunch of paths of the model $X^{(m)}_{T_0}, X^{(m)}_{T_1}, \ldots, X^{(m)}_{T_N}$ along with the corresponding payoff paths $Z^{(m)}_{T_0}, Z^{(m)}_{T_1}, \ldots, Z^{(m)}_{T_N}$, for $m = 1, \ldots, M$. To compute the τ_n's on each path, one needs to compute the conditional expectations $\mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]$ for $n = 1, \ldots, N-1$. Then, we introduce the final approximation of the backward iteration policy, in which the truncated expansion is computed using a Monte Carlo approximation

$$\begin{cases} \widehat{\tau}^{p,(m)}_N = T_N \\ \widehat{\tau}^{p,(m)}_n = T_n \mathbf{1}_{\left\{Z^{(m)}_{T_n} \geq \Phi_p(X^{(m)}_{T_n};\widehat{\theta}^{p,M}_n)\right\}} + \widehat{\tau}^{p,(m)}_{n+1} \mathbf{1}_{\left\{Z^{(m)}_{T_n} < \Phi^{(m)}_p(X^{(m)}_{T_n};\widehat{\theta}^{p,M}_n)\right\}}, \text{ for } 1 \leq n \leq N-1 \end{cases}$$

where $\widehat{\theta}^{p,M}_n$ solves the sample average approximation of (8)

$$\inf_{\theta\in\Theta_p} \frac{1}{M}\sum_{m=1}^{M}\left|\Phi_p(X^{(m)}_{T_n};\theta) - Z^{(m)}_{\tau^{p,(m)}_{n+1}}\right|^2. \tag{10}$$

Then, we finally approximate the time-0 price of the option by

$$U^{p,M}_0 = \max\left(Z_0, \frac{1}{M}\sum_{m=1}^{M} Z^{(m)}_{\widehat{\tau}^{p,(m)}_1}\right). \tag{11}$$

Remark 3.1 Note that to implement the previous algorithm we need to compute a minimizer for the optimization problem (10). Obviously this is not an easy task as this is a high-dimensional, non-convex and non smooth problem.

It is usually solved in practice using toolboxes as *Scikit-Learn* or *TensorFlow*, by means of a stochastic gradient descent method for which a full convergence proof under realistic assumptions are still unknown in our knowledge. See Bottou et al. [2018] or E et al. [2020] for recent in depth reviews of these subjects and Ghadimi and Lan [2013], Lei et al. [2019], Fehrman et al. [2020] for results for a non convex function.

4 Convergence of the algorithm

We start this section on the study of the convergence by introducing some bespoke notation following Clément et al. [2002].

4.1 Notation

First, it is important to note that the paths $\tau_1^{p,(m)}, \dots, \tau_N^{p,(m)}$ for $m = 1, \dots, M$ are identically distributed but not independent since the computations of θ_n^p at each time step n mix all the paths. We define the vector ϑ of the coefficients of the successive expansions $\vartheta^p = (\theta_1^p, \dots, \theta_{N-1}^p)$ and its Monte Carlo counterpart $\widehat{\vartheta}^{p,M} = (\widehat{\theta}_1^{p,M}, \dots, \widehat{\theta}_{N-1}^{p,M})$.

Now, we recall the notation used by Clément et al. [2002] to study the convergence of the original Longstaff Schwartz approach.

Given a deterministic parameter $t^p = (t_1^p, \dots, t_{N-1}^p)$ in ${\Theta_p}^{N-1}$ and deterministic vectors $z = z_1, \dots, z_N$ in $\mathbb{R}^N$ and $x = (x_1, \dots, x_N)$ in $(\mathbb{R}^r)^N$, we define the vector field $F = F_1, \dots, F_N$ by

$$\begin{cases} F_N(t^p, z, x) &= z_N \\ F_n(t^p, z, x) &= z_n \mathbf{1}_{\left\{z_n \geq \Phi_p(x_n; t_n^p)\right\}} + F_{n+1}(t^p, z, x) \mathbf{1}_{\left\{z_n < \Phi_p(x_n; t_n^p)\right\}}, \text{ for } 1 \leq n \leq N-1. \end{cases}$$

Note that $F_n(t, z, x)$ does not depend on the first $n-1$ components of t^p, ie $F_n(t^p, z, x)$ depends only $t_n^p, \dots, t_{N-1}^p$. Moreover,

$$\begin{aligned} F_n(\vartheta^p, Z, X) &= Z_{\tau_n^p}, \\ F_n(\widehat{\vartheta}^{p,M}, Z^{(m)}, X^{(m)}) &= Z^{(m)}_{\widehat{\tau}_n^{p,(m)}}. \end{aligned}$$

Moreover, we clearly have that for all $t^p \in {\Theta_p}^{N-1}$

$$|F_n(t^p, Z, X)| \leq \max_{k \geq n} |Z_{T_k}|. \tag{12}$$

4.2 Deep neural network approximations of conditional expectations

Proposition 4.1 *Assume that* $\mathbb{E}[\max_{0 \leq n \leq N} |Z_{T_n}|^2] < \infty$. *Then,* $\lim_{p \to \infty} \mathbb{E}[Z_{\tau_n^p} | \mathcal{F}_{T_n}] = \mathbb{E}[Z_{\tau_n} | \mathcal{F}_{T_n}]$ *in* $L^2(\Omega)$ *for all* $1 \leq n \leq N$.

Remark 4.2 Note that in the proof of Proposition 4.1, there is no need for the sets Θ_p to be compact for every p. We could have chosen $\gamma_p = \infty$. However, the boundedness assumption will be required in the following section, so to work with the same approximations over the whole paper, we have decided to impose compactness on Θ_p for every p.

Proof. q We proceed by induction. The result is true for $n = N$ as $\tau_N = \tau_N^p = T$. Assume it holds for $n+1$ (with $0 \leq n \leq N-1$), we will prove it is true for n. For this, using both recursion equations, we have

$$\begin{aligned} \mathbb{E}[Z_{\tau_n^p} - Z_{\tau_n} | \mathcal{F}_{T_n}] = Z_{T_n} &\left(\mathbf{1}_{\left\{Z_{T_n} \geq \Phi_p(X_{T_n}; \theta_n^p)\right\}} - \mathbf{1}_{\left\{Z_{T_n} \geq \mathbb{E}[Z_{\tau_{n+1}} | \mathcal{F}_{T_n}]\right\}} \right) \\ &+ \mathbb{E}\left[Z_{\tau_{n+1}^p} \mathbf{1}_{\left\{Z_{T_n} < \Phi_p(X_{T_n}; \theta_n^p)\right\}} - Z_{\tau_{n+1}} \mathbf{1}_{\left\{Z_{T_n} < \mathbb{E}[Z_{\tau_{n+1}} | \mathcal{F}_{T_n}]\right\}} | \mathcal{F}_{T_n} \right]. \end{aligned}$$

Now, defining A_n^p as

$$A_n^p = (Z_{T_n} - \mathbb{E}[Z_{\tau_{n+1}} | \mathcal{F}_{T_n}]) \left(\mathbf{1}_{\left\{Z_{T_n} \geq \Phi_p(X_{T_n}; \theta_n^p)\right\}} - \mathbf{1}_{\left\{Z_{T_n} \geq \mathbb{E}[Z_{\tau_{n+1}} | \mathcal{F}_{T_n}]\right\}} \right),$$

we obtain

$$\mathbb{E}[Z_{\tau_n^p} - Z_{\tau_n}|\mathcal{F}_{T_n}] = A_n^p + \mathbb{E}\left[Z_{\tau_{n+1}^p} - Z_{\tau_{n+1}}|\mathcal{F}_{T_n}\right]\mathbf{1}_{\left\{Z_{T_n} < \Phi_p(X_{T_n};\theta_n^p)\right\}}. \tag{13}$$

By the induction assumption, the term $\mathbb{E}\left[Z_{\tau_{n+1}^p} - Z_{\tau_{n+1}}|\mathcal{F}_{T_n}\right]$ goes to zero in $L^2(\Omega)$ as p goes to infinity. So, we just have to prove that A_n^p converges to zero in $L^2(\Omega)$ when $p \to \infty$. For this, note that

$$\begin{aligned}
|A_n^p| &\leq \left|Z_{T_n} - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right| \left|\mathbf{1}_{\left\{Z_{T_n} \geq \Phi_p(X_{T_n};\theta_n^p)\right\}} - \mathbf{1}_{\left\{Z_{T_n} \geq \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right\}}\right| \\
&\leq \left|Z_{T_n} - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right| \left|\mathbf{1}_{\left\{\mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}] > Z_{T_n} \geq \Phi_p(X_{T_n};\theta_n^p)\right\}} - \mathbf{1}_{\left\{\Phi_p(X_{T_n};\theta_n^p) > Z_{T_n} \geq \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right\}}\right| \\
&\leq \left|Z_{T_n} - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right| \mathbf{1}_{\left\{\left|Z_{T_n} - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right| \leq \left|\Phi_p(X_{T_n};\theta_n^p) - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right|\right\}} \\
&\leq \left|\Phi_p(X_{T_n};\theta_n^p) - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right|.
\end{aligned}$$

So we obtain

$$|A_n^p| \leq \left|\Phi_p(X_{T_n};\theta_n^p) - \mathbb{E}[Z_{\tau_{n+1}^p}|\mathcal{F}_{T_n}]\right| + \left|\mathbb{E}[Z_{\tau_{n+1}^p}|\mathcal{F}_{T_n}] - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right|. \tag{14}$$

Morevoer, as the conditional expectation is an orthogonal projection, we clearly have that

$$\mathbb{E}\left[\left|\mathbb{E}[Z_{\tau_{n+1}^p}|\mathcal{F}_{T_n}] - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right|^2\right] \leq \mathbb{E}\left[\left|\mathbb{E}[Z_{\tau_{n+1}^p}|\mathcal{F}_{T_{n+1}}] - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_{n+1}}]\right|^2\right]. \tag{15}$$

Then, the induction assumption for $n+1$ yields that the second term on the r.h.s of (14) goes to zero in $L^2(\Omega)$ when $p \to \infty$.

To deal with the first term on the r.h.s of (14), we introduce for any $p \in \mathbb{N}$, $\tilde{\theta}_n^p \in \Theta_p$ defined as a minimiser to

$$\inf_{\theta \in \Theta_p} \mathbb{E}\left[\left|\Phi_p(X_{T_n};\theta) - \mathbb{E}\left[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}\right]\right|^2\right]. \tag{16}$$

Note that $\Phi(\cdot, \tilde{\theta}_n^p)$ is the best approximation on $\mathcal{NN}_p$ of the true continuation value at time n. As θ_n^p solves (9), we clearly have that

$$\begin{aligned}
&\mathbb{E}\left[\left|\Phi_p(X_{T_n};\theta_n^p) - \mathbb{E}[Z_{\tau_{n+1}^p}|\mathcal{F}_{T_n}]\right|^2\right] \\
&\leq \mathbb{E}\left[\left|\Phi_p(X_{T_n};\tilde{\theta}_n^p) - \mathbb{E}[Z_{\tau_{n+1}^p}|\mathcal{F}_{T_n}]\right|^2\right] \\
&\leq 2\mathbb{E}\left[\left|\Phi_p(X_{T_n};\tilde{\theta}_n^p) - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right|^2\right] + 2\mathbb{E}\left[\left|\mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}] - \mathbb{E}[Z_{\tau_{n+1}^p}|\mathcal{F}_{T_n}]\right|^2\right]
\end{aligned} \tag{17}$$

Using the induction assumption for $n+1$, the second term on the r.h.s of (17) goes to zero in $L^2(\Omega)$ and from the universal approximation theorem (see Theorem 2.2 and Remark (2.3)), we deduce that

$$\lim_{p\to\infty} \mathbb{E}\left[\left|\Phi_p(X_{T_n};\tilde{\theta}_n^p) - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}]\right|^2\right] = 0.$$

Then, we conclude that $\lim_{p\to\infty} \mathbb{E}[|A_n^p|^2] = 0$. ■

The next proposition show that if we have an estimate of the speed of convergence for the network approximation for a suitable class of functions we are able to derive the speed of convergence for the Bermudan option price.

Proposition 4.3 *Assume that for every* $0 \leq n \leq N-1$*, there exists a sequence* $(\delta_n^p)_p$ *of positive real numbers such that*

$$\inf_{\theta \in \Theta_p} \mathbb{E}\left[\left|\Phi_p(X_{T_n};\theta) - \mathbb{E}\left[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}\right]\right|^2\right] = O(\delta_n^p) \quad \textit{when } p \to \infty. \tag{18}$$

Then,

$$\mathbb{E}\left[|\mathbb{E}[Z_{\tau_n^p}|\mathcal{F}_{T_n}] - \mathbb{E}[Z_{\tau_n}|\mathcal{F}_{T_n}]|^2\right] = O\left(\sum_{i=n}^{N-1} \delta_i^p\right) \quad \textit{when } p \to \infty.$$

Proof. We use the same notation as in the proof of Proposition 4.1. We proceed by backward induction.

First note that $\tau_N^p = \tau_N = T$ and then, using (13), $A_{N-1}^p = \mathbb{E}[Z_{\tau_{N-1}}^p - Z_{\tau_{N-1}}|\mathcal{F}_{T_{N-1}}]$. Moreover, from (14), we get that

$$\left|A_{N-1}^p\right| \leq \left|\Phi_p(X_{T_{N-1}};\theta_n^p) - \mathbb{E}[Z_{\tau_N^p}|\mathcal{F}_{T_{N-1}}]\right|.$$

Using again that $\tau_N^p = \tau_N$, we deduce that

$$\mathbb{E}\left[\left|\mathbb{E}[Z_{\tau_{N-1}^p}|\mathcal{F}_{T_{N-1}}] - \mathbb{E}[Z_{\tau_{N-1}}|\mathcal{F}_{T_{N-1}}]\right|^2\right] \leq \inf_{\theta \in \Theta_p} \mathbb{E}\left[\left|\Phi_p(X_{T_{N-1}};\theta) - \mathbb{E}\left[Z_{\tau_N}|\mathcal{F}_{T_{N-1}}\right]\right|^2\right].$$

Therefore,

$$\mathbb{E}\left[\left|\mathbb{E}[Z_{\tau_{N-1}^p}|\mathcal{F}_{T_{N-1}}] - \mathbb{E}[Z_{\tau_{N-1}}|\mathcal{F}_{T_{N-1}}]\right|^2\right] = O(\delta_{N-1}^p) \quad \text{when} \quad p \to \infty.$$

Assume the result holds true for $n+1$ (with $0 \leq n \leq N-2$), we will prove it is true for n.

$$\begin{aligned}
\mathbb{E}\left[|\mathbb{E}[Z_{\tau_n^p} - Z_{\tau_n}|\mathcal{F}_{T_n}]|^2\right] &\leq 2\mathbb{E}\left[\left|\mathbb{E}\left[Z_{\tau_{n+1}^p} - Z_{\tau_{n+1}}|\mathcal{F}_{T_n}\right]\right|^2\right] + 2\mathbb{E}[|A_n^p|^2] \\
&\leq 2\mathbb{E}\left[\left|\mathbb{E}\left[Z_{\tau_{n+1}^p} - Z_{\tau_{n+1}}|\mathcal{F}_{T_{n+1}}\right]\right|^2\right] + 2\mathbb{E}[|A_n^p|^2]
\end{aligned}$$

where we have used (15). Then, using the induction assumption, we get

$$\mathbb{E}\left[|\mathbb{E}[Z_{\tau_n^p} - Z_{\tau_n}|\mathcal{F}_{T_n}]|^2\right] \leq O(\delta_{n+1}^p) + 2\mathbb{E}[|A_n^p|^2].$$

From (14) and (15), we have

$$\begin{aligned}
\mathbb{E}[|A_n^p|^2] &\leq 2\mathbb{E}\left[\left|\Phi_p(X_{T_n};\theta_n^p) - \mathbb{E}[Z_{\tau_{n+1}^p}|\mathcal{F}_{T_n}]\right|^2\right] + 2\mathbb{E}\left[\left|\mathbb{E}[Z_{\tau_{n+1}^p}|\mathcal{F}_{T_{n+1}}] - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_{n+1}}]\right|^2\right] \\
&\leq 4\inf_{\theta \in \Theta_p} \mathbb{E}\left[\left|\Phi_p(X_{T_n};\theta) - \mathbb{E}\left[Z_{\tau_{n+1}}|\mathcal{F}_{T_n}\right]\right|^2\right] \\
&\quad + 6\mathbb{E}\left[\left|\mathbb{E}[Z_{\tau_{n+1}^p}|\mathcal{F}_{T_{n+1}}] - \mathbb{E}[Z_{\tau_{n+1}}|\mathcal{F}_{T_{n+1}}]\right|^2\right]
\end{aligned}$$

where the last inequality comes from (17).

From the induction assumption, the second term is bounded by $O\left(\sum_{i=n+1}^{N-1} \delta_i^p\right)$. From (18), the first term is bounded by $O(\delta_n^p)$. Then, we conclude that when $p \to \infty$

$$E\left[\left|\mathbb{E}[Z_{\tau_n^p}|\mathcal{F}_{T_n}] - \mathbb{E}[Z_{\tau_n}|\mathcal{F}_{T_n}]\right|^2\right] = O\left(\sum_{i=n}^{N-1} \delta_i^p\right).$$

■

4.3 Convergence of the Monte Carlo approximation

In the following, we assume that p is fixed and we study the convergence with respect to the number of samples M. First, we recall some important results on the convergence of the solution of a sequence of optimization problems whose cost functions converge.

4.3.1 Convergence of optimization problems

Consider a sequence of real valued functions $(f_n)_n$ defined on a compact set $K \subset \mathbb{R}^d$. Define,

$$v_n = \inf_{x \in K} f_n(x)$$

and let x_n be a sequence of minimizers

$$f_n(x_n) = \inf_{x \in K} f_n(x).$$

From [Rubinstein and Shapiro, 1993, Chap. 2], we have the following result.

Lemma 4.4 *Assume that the sequence $(f_n)_n$ converges uniformly on K to a continuous function f. Let $v^\star = \inf_{x \in K} f(x)$ and $\mathcal{S}^\star = \{x \in K : f(x) = v^\star\}$. Then $v_n \to v^\star$ and $d(x_n, \mathcal{S}^\star) \to 0$ a.s.*

In the following, we will also make heavy use of the following result, which is a restatement of the law of large numbers in Banach spaces, see [Ledoux and Talagrand, 1991, Corollary 7.10, page 189] or [Rubinstein and Shapiro, 1993, Lemma A1].

Lemma 4.5 *Let $(\xi_i)_{i\geq 1}$ be a sequence of i.i.d. $\mathbb{R}^m$-valued random vectors and $h : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ be a measurable function. Assume that*

- *a.s., $\theta \in \mathbb{R}^d \mapsto h(\theta, \xi_1)$ is continuous,*
- $\forall C > 0,\ \mathbb{E}\left[\sup_{|\theta|\leq C} |h(\theta, \xi_1)|\right] < +\infty.$

Then, a.s. $\theta \in \mathbb{R}^d \mapsto \frac{1}{n}\sum_{i=1}^n h(\theta, \xi_i)$ converges locally uniformly to the continuous function $\theta \in \mathbb{R}^d \mapsto \mathbb{E}[h(\theta, \xi_1)]$, ie

$$\lim_{n\to\infty} \sup_{|\theta|\leq C} \left|\frac{1}{n}\sum_{i=1}^n h(\theta, \xi_i) - \mathbb{E}[h(\theta, \xi_1)]\right| = 0\, a.s.$$

4.3.2 Strong law of large numbers

To prove a strong law of large numbers, we will need the following assumptions.

($\mathcal{H}$-1) For every $p \in \mathbb{N}$, $p > 1$, there exist $q \geq 1$ and $\kappa_p > 0$ s.t.

$$\forall\, x \in \mathbb{R}^r,\ \forall\, \theta \in \Theta_p, \quad |\Phi_p(x, \theta)| \leq \kappa_p(1 + |x|^q).$$

Moreover, for all $1 \leq n \leq N - 1$, a.s. the random functions $\theta \in \Theta_p \longmapsto \Phi_p(X_{T_n}, \theta)$ are continuous. Note that as Θ_p is a compact set, the continuity automatically yields the uniform continuity.

($\mathcal{H}$-2) For q defined in ($\mathcal{H}$-1), $\mathbb{E}[|X_{T_n}|^{2q}] < \infty$ for all $0 \leq n \leq N$.

($\mathcal{H}$-3) For all $p \in \mathbb{N}$, $p > 1$ and all $1 \leq n \leq N - 1$, $\mathbb{P}\left(Z_{T_n} = \Phi_p(X_{T_n}; \theta_n^p)\right) = 0$.

We introduce the notation

$$\mathcal{S}_n^p = \arg \inf_{\theta \in \Theta_p} \mathbb{E}\left[\left|\Phi_p(X_{T_n}; \theta) - Z_{\tau_{n+1}^p}\right|^2\right]. \tag{19}$$

Note that $\mathcal{S}_n^p$ is a non void compact set.

($\mathcal{H}$-4) For every $p \in \mathbb{N}$, $p > 1$ and every $1 \leq n \leq N$, for all $\theta^1, \theta^2 \in \mathcal{S}_n^p$,

$$\Phi_p(x; \theta^1) = \Phi_p(x; \theta^2) \quad \text{for all } x \in \mathbb{R}^r$$

Remark 4.6 Assumption ($\mathcal{H}$-1) is clearly satisfied for the classical activation functions ReLU $\sigma_a(x) = (x)_+$, sigmoid $\sigma_a(x) = (1 + \mathrm{e}^{-x})^{-1}$ and $\sigma_a(x) = \tanh(x)$. When the law of X_{T_n} has a density with respect to the Lebesgue measure, the continuity assumption stated in ($\mathcal{H}$-1) is even satisfied by the binary step activation function $\sigma_a(x) = \mathbf{1}_{\{x \geq 0\}}$.

Remark 4.7 Considering the natural symmetries existing in a neural network, it is clear that the set $\mathcal{S}_n^p$ will hardly ever be reduced to a singleton. So, none of the parameters $\widehat{\theta}_n^{p,M}$ or θ_n^p is unique. Here, we only require the function described by the neural network approximation to be unique but not its representation, which is much weaker and more realistic in practice. We refer to Albertini et al. [1993], Albertini and Sontag [1994] for characterization of symmetries of neural networks and to Williamson and Helmke [1995] for results on existence and uniqueness of an optimal neural network approximation (but not its parameters).

To start, we prove the convergence of the neural network approximation of the conditional expectation at each time step.

Proposition 4.8 *Assume that Assumptions ($\mathcal{H}$-1)-($\mathcal{H}$-4) hold. Let θ_n^p be a minimiser of optimization problem (19) and $\widehat{\theta}_n^{p,M}$ be a minimiser of the sample average optimization problem (10), then, for every $n = 1, \ldots, N$, $\Phi_p(X_{T_n}^{(1)}; \widehat{\theta}_n^{p,M})$ converges to $\Phi_p(X_{T_n}^{(1)}; \theta_n^p)$ a.s. as $M \to \infty$.*

Lemma 4.9 *For every $n = 1, \dots, N-1$,*

$$|F_n(a, Z, X) - F_n(b, Z, X)| \leq \left(\sum_{i=n}^{N} |Z_{T_i}|\right) \left(\sum_{i=n}^{N-1} \mathbf{1}_{\left\{\left|Z_{T_i} - \Phi_p(X_{T_i}; b_i)\right| \leq \left|\Phi_p(X_{T_i}; a_i) - \Phi_p(X_{T_i}; b_i)\right|\right\}}\right)$$

Proof (Proof of Proposition 4.8). We proceed by induction.
Step 1. For $n = N-1$, $\widehat{\theta}^{p,M}_{N-1}$ solves

$$\inf_{\theta \in \Theta_p} \sum_{m=1}^{M} \left|\Phi_p(X^{(m)}_{T_{N-1}}; \theta) - Z^{(m)}_{T_N}\right|^2 .$$

We aim at applying Lemma 4.5 to the sequence of i.i.d. random functions $h_m(\theta) = \left|\Phi_p(X^{(m)}_{T_{N-1}}; \theta) - Z^{(m)}_{T_N}\right|^2$. From Assumptions ($\mathcal{H}$-1) and ($\mathcal{H}$-2), we deduce that

$$\mathbb{E}\left[\sup_{\theta \in \Theta_P} |h_m(\theta)|\right] \leq 2\kappa_p \mathbb{E}[\left|X_{T_{N-1}}\right|^{2q}] + \mathbb{E}[(Z_T)^2] < \infty.$$

Then, Lemma 4.5 implies that a.s. the function

$$\theta \in \Theta_p \longmapsto \frac{1}{M} \sum_{m=1}^{M} \left|\Phi_p(X^{(m)}_{T_{N-1}}; \theta) - Z^{(m)}_{T_N}\right|^2$$

converges uniformly to $\mathbb{E}[\left|\Phi_p(X_{T_{N-1}}; \theta) - Z_{T_N}\right|^2]$. Hence, we deduce from Lemma 4.4 that $d(\widehat{\theta}^{p,M}_{N-1}, S^p_{N-1}) \to 0$ a.s. when $M \to \infty$. We restrict to a subset with probability one of the original probability space on which this convergence holds and the random functions $\Phi(X^{(1)}_{T_{N-1}}; \cdot)$ are uniformly continuous, see ($\mathcal{H}$-1). There exists a sequence $(\xi^{p,M}_{N-1})_M$ taking values in S^p_{N-1} such that $\left|\widehat{\theta}^{p,M}_{N-1} - \xi^{p,M}_{N-1}\right| \to 0$, when $M \to \infty$. The uniform continuity of the random functions $\Phi(X^{(1)}_{T_{N-1}}; \cdot)$ yields that

$$\Phi(X^{(1)}_{T_{N-1}}; \widehat{\theta}^{p,M}_{N-1}) - \Phi(X^{(1)}_{T_{N-1}}; \xi^{p,M}_{N-1}) \to 0$$

Then, we conclude from Assumption ($\mathcal{H}$-4), that $\Phi(X^{(1)}_{T_{N-1}}; \widehat{\theta}^{p,M}_{N-1}) \to \Phi(X^{(1)}_{T_{N-1}}; \theta^p_{N-1})$

Step 2. Choose $n \leq N-2$ and assume that the convergence result holds for $n+1, \dots, N-1$, we aim at proving this is true for n. We recall that $\widehat{\theta}^{p,M}_n$ solves

$$\inf_{\theta \in \Theta_p} \sum_{m=1}^{M} \left|\Phi_p(X^{(m)}_{T_n}; \theta) - F_{n+1}(\widehat{\vartheta}^{p,M}, Z^{(m)}, X^{(m)})\right|^2 .$$

We introduce the two random functions for $\theta \in \Theta_p$

$$\hat{v}^M(\theta) = \frac{1}{M}\sum_{m=1}^{M}\left|\Phi_p(X_{T_n}^{(m)};\theta) - F_{n+1}(\widehat{\vartheta}^{p,M}, Z^{(m)}, X^{(m)})\right|^2$$

$$v^M(\theta) = \frac{1}{M}\sum_{m=1}^{M}\left|\Phi_p(X_{T_n}^{(m)};\theta) - F_{n+1}(\vartheta^{p}, Z^{(m)}, X^{(m)})\right|^2.$$

The function v^M clearly writes as the sum of i.i.d. random variables. Moreover, by combining (12) and Assumptions ($\mathcal{H}$-1) and ($\mathcal{H}$-2), we obtain

$$\mathbb{E}\left[\sup_{\theta\in\Theta_p}|\Phi_p(X_{T_n};\theta) - F_{n+1}(\vartheta^p, Z, X)|^2\right] \le 2\kappa\mathbb{E}[1+|X_{T_n}|^{2q}] + \mathbb{E}\left[\max_{\ell\ge n+1}(Z_{T_\ell})^2\right] < \infty.$$

Then, the sequence of random functions v^M a.s. converges uniformly to the continuous function v defined for $\theta \in \Theta_p$ by

$$v(\theta) = \mathbb{E}\left[|\Phi_p(X_{T_n};\theta) - F_{n+1}(\vartheta^p, Z, X)|^2\right].$$

It remains to prove that $\sup_{\theta\in\Theta_p}\left|\hat{v}^M(\theta) - v^M(\theta)\right| \to 0$ a.s. when $M \to \infty$.

$$\begin{aligned}
&\left|\hat{v}^M(\theta) - v^M(\theta)\right| \\
&\le \frac{1}{M}\sum_{m=1}^{M}\left|2\Phi_p(X_{T_n}^{(m)};\theta) - F_{n+1}(\widehat{\vartheta}^{p,M}, Z^{(m)}, X^{(m)}) - F_{n+1}(\vartheta^{p}, Z^{(m)}, X^{(m)})\right| \\
&\qquad\qquad \left|F_{n+1}(\widehat{\vartheta}^{p,M}, Z^{(m)}, X^{(m)}) - F_{n+1}(\vartheta^{p}, Z^{(m)}, X^{(m)})\right| \\
&\le \frac{1}{M}\sum_{m=1}^{M} 2\left(\kappa(1+|X_{T_n}^{(m)}|^q) + \max_{\ell\ge n+1}|Z_{T_\ell}|\right) \\
&\qquad\qquad \left|F_{n+1}(\widehat{\vartheta}^{p,M}, Z^{(m)}, X^{(m)}) - F_{n+1}(\vartheta^{p}, Z^{(m)}, X^{(m)})\right|
\end{aligned}$$

where we have used (12) and Assumptions ($\mathcal{H}$-1) and ($\mathcal{H}$-2). Then from Lemma 4.9, we can write

$$\begin{aligned}
&\left|\hat{v}^M(\theta) - v^M(\theta)\right| \\
&\le \frac{1}{M}\sum_{m=1}^{M} 2\left(\kappa(1+|X_{T_n}^{(m)}|^q) + \max_{\ell\ge k+1}|Z_{T_\ell}|\right) \\
&\qquad\qquad \left(\sum_{i=n+1}^{N}\left|Z_{T_i}^{(m)}\right|\right)\left(\sum_{i=n+1}^{N-1}\mathbf{1}_{\left\{\left|Z_{T_i}^{(m)} - \Phi_p(X_{T_i}^{(m)};\theta_i^p)\right| \le \left|\Phi_p(X_{T_i}^{(m)};\widehat{\theta}_i^{p,M}) - \Phi_p(X_{T_i}^{(m)};\theta_i^p)\right|\right\}}\right) \\
&\le \frac{1}{M}\sum_{m=1}^{M} C\left(\kappa_p(1+|X_{T_n}^{(m)}|^{2q}) + \sum_{i=n+1}^{N}\left|Z_{T_i}^{(m)}\right|^2\right) \\
&\qquad\qquad \left(\sum_{i=n+1}^{N-1}\mathbf{1}_{\left\{\left|Z_{T_i}^{(m)} - \Phi_p(X_{T_i}^{(m)};\theta_i^p)\right| \le \left|\Phi_p(X_{T_i}^{(m)};\widehat{\theta}_i^{p,M}) - \Phi_p(X_{T_i}^{(m)};\theta_i^p)\right|\right\}}\right)
\end{aligned}$$

where C is a generic constant only depending on κ_p, n and N.

Let $\varepsilon > 0$. Using the induction assumption and the strong law of large numbers, we have

$$\begin{aligned}
&\limsup_M \sup_{\theta \in \Theta_p} \left| \hat{v}^M(\theta) - v^M(\theta) \right| \\
&\leq \limsup_M \frac{1}{M} \sum_{m=1}^M C \left((1 + |X_{T_n}^{(m)}|^{2q}) + \sum_{i=n+1}^N \left| Z_{T_i}^{(m)} \right|^2 \right) \\
&\qquad\qquad \left(\sum_{i=n+1}^{N-1} \mathbf{1}_{\left\{ \left| Z_{T_i}^{(m)} - \Phi_p(X_{T_i}^{(m)};\theta_i^p) \right| \leq \varepsilon \right\}} \right) \\
&\leq C \mathbb{E} \left[\left((1 + |X_{T_n}|^{2q}) + \sum_{i=n+1}^N |Z_{T_i}|^2 \right) \left(\sum_{i=n+1}^{N-1} \mathbf{1}_{\left\{ \left| Z_{T_i} - \Phi_p(X_{T_i};\theta_i^p) \right| \leq \varepsilon \right\}} \right) \right]
\end{aligned}$$

From ($\mathcal{H}$-3), we deduce that $\lim_{\varepsilon \to 0} \mathbf{1}_{\left\{ \left| Z_{T_i} - \Phi_p(X_{T_i};\theta_i^p) \right| \leq \varepsilon \right\}} = 0$ a.s. and we conclude that a.s. $\hat{v}^M - v^M$ converges to zero uniformly. As we have already proved that a.s. v^M converges uniformly to the continuous function v, we deduce that a.s. $\hat{v}^M$ converges uniformly to v. From Lemma 4.4, we conclude that $d(\widehat{\theta}_n^{p,M}, S_n^p) \to 0$ a.s. when $M \to \infty$. We restrict to a subset with probability one of the original probability space on which this convergence holds and the random functions $\Phi(X_{T_{N-1}}^{(1)}; \cdot)$ are uniformly continuous, see ($\mathcal{H}$-1). There exists a sequence $(\xi_n^{p,M})_M$ taking values in S_n^p such that $\left| \widehat{\theta}_n^{p,M} - \xi_n^{p,M} \right| \to 0$ when $M \to \infty$. The uniform continuity of the random functions $\Phi(X_{T_{N-1}}^{(1)}; \cdot)$ yields that

$$\Phi(X_{T_{N-1}}^{(1)}; \widehat{\theta}_n^{p,M}) - \Phi(X_{T_n}^{(1)}; \xi_n^{p,M}) \to 0 \quad \text{when } M \to \infty.$$

Then, we conclude from Assumption ($\mathcal{H}$-4), that $\Phi(X_{T_n}^{(1)}; \widehat{\theta}_n^{p,M}) \to \Phi(X_{T_n}^{(1)}; \theta_n^p)$ when $M \to \infty$. ■

Now that the convergence of the expansion is established, we can study the convergence of $U_0^{p,M}$ to U_0^p when $M \to \infty$.

Theorem 4.10 *Assume that Assumptions ($\mathcal{H}$-1)-($\mathcal{H}$-4) hold. Then, for $\alpha = 1, 2$ and every $n = 1, \ldots, N$,*

$$\lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^M \left(Z_{\widehat{\tau}_n^{p,(m)}}^{(m)} \right)^\alpha = \mathbb{E}\left[(Z_{\tau_n^p})^\alpha \right] \quad a.s.$$

Proof. Note that $\mathbb{E}[(Z_{\tau_n^p})^\alpha] = \mathbb{E}[F_n(\vartheta^p, Z, X)^\alpha]$ and by the strong law of large numbers

$$\lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^M F_n(\vartheta^p, Z^{(m)}, X^{(m)})^\alpha = \mathbb{E}[F_n(\vartheta^p, Z, X)^\alpha] \quad a.s.$$

Hence, we have to prove that

$$\Delta F_M = \frac{1}{M} \sum_{m=1}^M \left(F_n(\widehat{\vartheta}^{p,M}, Z^{(m)}, X^{(m)})^\alpha - F_n(\vartheta^p, Z^{(m)}, X^{(m)})^\alpha \right) \xrightarrow[M \to \infty]{a.s} 0.$$

For any $x, y \in \mathbb{R}$, and $\alpha = 1, 2$, $|x^\alpha - y^\alpha| = |x - y|\,|x^{q-1} + y^{q-1}|$. Using Lemma 4.9 and that $|F_n(\gamma, z, g)| \leq \max_{n \leq j \leq N} |z_j|$, we have

$$
\begin{aligned}
|\Delta F_M| &\leq \frac{1}{M} \sum_{m=1}^{M} \left| F_n(\widehat{\vartheta}^{p,M}, Z^{(m)}, X^{(m)})^\alpha - F_n(\vartheta^p, Z^{(m)}, G^{(m)})^\alpha \right| \\
&\leq 2\, \frac{1}{M} \sum_{m=1}^{M} \sum_{i=n}^{N} \max_{n \leq j \leq N} \left| Z_{T_j}^{(m)} \right| \left| Z_{T_{i+1}}^{(m)} \right| \left(\sum_{i=n}^{N-1} \mathbf{1}_{\left\{ \left| Z_{T_i}^{(m)} - \Phi_p(X_{T_i}^{(m)}; \theta_i^p) \right| \leq \left| \Phi_p(X_{T_i}^{(m)}; \widehat{\theta}_i^{p,M}) - \Phi_p(X_{T_i}^{(m)}; \theta_i^p) \right| \right\}} \right)
\end{aligned}
$$

Using Proposition 4.8, for all $i = n, \dots, N-1$, $\left| \Phi_p(X_{T_i}^{(1)}; \widehat{\theta}_i^{p,M}) - \Phi_p(X_{T_i}^{(1)}; \theta_i^p) \right| \to 0$ when $M \to \infty$. Then for any $\varepsilon > 0$,

$$
\begin{aligned}
&\limsup_M |\Delta F_M| \\
&\leq 2 \limsup_M \frac{1}{M} \sum_{m=1}^{M} \sum_{i=n}^{N} \max_{n \leq j \leq N} \left| Z_{T_j}^{(m)} \right| \left| Z_{T_{i+1}}^{(m)} \right| \left(\sum_{i=k}^{N-1} \mathbf{1}_{\left\{ \left| Z_{T_i}^{(m)} - \Phi_p(X_{T_i}^{(m)}; \theta_i^p) \right| \leq \varepsilon \right\}} \right) \\
&\leq 2\mathbb{E} \left[\sum_{i=n}^{N} \max_{n \leq j \leq N} \left| Z_{T_j} \right| \left| Z_{T_{i+1}} \right| \left(\sum_{i=n}^{N-1} \mathbf{1}_{\left\{ \left| Z_{T_i} - \Phi_p(X_{T_i}^{(m)}; \theta_i^p) \right| \leq \varepsilon \right\}} \right) \right]
\end{aligned}
$$

where the last inequality follows from the strong law of larger numbers as $\mathbb{E}[\max_{n \leq j \leq N} \left| Z_{T_j} \right|^2] < \infty$. We conclude that $\limsup_M |\Delta F_M| = 0$ by letting ε go to 0 and by using ($\mathcal{H}$-3). ■

The case $\alpha = 1$ proves the strong law of large numbers for the algorithm. Note that solving the minimisation problem (10) mixes all stopped paths $Z^{(m)}_{\widehat{\tau}_n^{p,(m)}}$, it is unlikely that the estimators $\frac{1}{M} \sum_{m=1}^{M} Z^{(m)}_{\widehat{\tau}_n^{p,(m)}}$ for $1 \leq n \leq N$ are unbiased. We recall that $U_n^{p,M} = \frac{1}{M} \sum_{m=1}^{M} F_n(\widehat{\vartheta}^{p,M}, Z^{(m)}, G^{(m)})$ and $Z_{\tau_n^p} = F_n(\vartheta^p, Z, X)$. Then,

$$
\begin{aligned}
\mathbb{E}\left[U_n^{p,M}\right] - \mathbb{E}\left[Z_{\tau_n^p}\right] &= \mathbb{E}\left[\frac{1}{M} \sum_{m=1}^{M} \left(F_n(\widehat{\vartheta}^{p,M}, Z^{(m)}, X^{(m)}) - F_n(\vartheta^p, Z^{(m)}, X^{(m)}) \right) \right] \\
&= \mathbb{E}\left[F_n(\widehat{\vartheta}^{p,M}, Z^{(1)}, X^{(1)}) - F_n(\vartheta^p, Z^{(1)}, X^{(1)}) \right]
\end{aligned}
$$

where we have used that all the random variables have the same distribution.

5 Numerical experiments

In this section, we compare the results given by the standard Longstaff Schwartz approach with polynomial regression to the algorithm described in Section 3. The only difference between the two methods lies in the way of approximating the conditional expectation at each time step. The two algorithms are implemented in Python using the *PolynomialFeatures* toolbox of *scikit-learn* (Pedregosa et al. [2011]) for the polynomial regression and the *tensorFlow* toolbox (Abadi et al. [2015]) to compute the neural network approximation. We have chosen options for which there is a substantial gap between the European and Bermudan prices, which means that there exists indeed an early exercise strategy and that the accuracy of the conditional expectations approximations plays a major role.

Details on the algorithm used in the experiments In all the experiments, we have run our algorithm 100 times to compute the average price along with the half-width of the confidence interval for the price estimator reported in the tables between parentheses in the form $(\pm\cdot)$. Although the confidence interval is informative to know how much we can trust a price, it completely squeezes the bias related to the approximation of the conditional expectations. Remember that the estimator given by (11) is not an unbiased estimator and one should therefore be very careful when comparing the results. Keep in mind that a higher price does not always mean a better price.

For the activation function σ_a in (5), we have used the leaky ReLU function defined by

$$\sigma_a(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0.3 \times x & \text{if } x < 0 \end{cases}$$

We relied on the ADAM algorithm to fit the neural network at each time step and the columns *epochs* refer to the number of times we go through the entire data set to train the network. Note that using *epochs* $= 1$ corresponds to the standard approach used in online stochastic approximation, in which each data is used only once. We use the same neural network through all the time steps and in particular at a time step $0 < n < N-1$, we take the optimal parameter at time $n+1$, $\widehat{\theta}^{p,M}_{n+1}$, as the starting point of the training algorithm. Because of this smart choice, there is actually no use setting *epochs*> 1 for $n < N-1$. We observed in our numerical experiments that passing over all the data several times does not reduce the training error at times $0 < n < N-1$, whereas it does help when fitting the first neural network at time $N-1$. This allows for huge computational time savings.

For learning the continuation value at each exercising date, we only use the in-the-money paths as already suggested in the original Longstaff Schwartz algorithm Longstaff and Schwartz [2001]. This means that the definition of the optimization problem (8) has to be changed into

$$\inf_{\theta \in \Theta_p} \mathbb{E}\left[\left| \Phi_p(X_{T_n}; \theta) - Z_{\tau^p_{n+1}} \right|^2 \mathbf{1}_{\{Z_{T_n} > 0\}} \right].$$

The empirical counterpart (10) needs to be adapted in a similar way. Note that it does not change the theoretical analysis of the algorithm but it is numerically more efficient. We proceed similarly in the original Longstaff Schwartz algorithm we are comparing to in the next sections.

5.1 Examples in the Black Scholes model

The $d-$dimensional Black Scholes model writes for $j \in \{1, \dots, d\}$

$$dS_t^j = S_t^j (r_t dt + \sigma^j L_j dB_t)$$

where B is a Brownian motion with values in $\mathbb{R}^d$, $\sigma = (\sigma^1, \dots, \sigma^d)$ is the vector of volatilities, assumed to be deterministic and positive at all times and L_j is the j-th row of the matrix L defined as a square root of the correlation matrix Γ, given by

$$\Gamma = \begin{pmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho \\ \rho & \dots & \rho & 1 \end{pmatrix}$$

where $\rho \in]-1/(d-1), 1]$ to ensure that Γ is positive definite.

5.1.1 Benchmarking the method on the one-dimensional put option

Before investigating more elaborate numerical examples, we want to test our method on the one dimensional put option. As standard as this example might be, getting a trustworthy reference price is not an easy task. For this example, we compare our approach to the benchmark price computed by a convolution method in Lord et al. [2008] and later used as a reference price in Fang and Oosterlee [2009]. Their reference price is 11.987 where all the digits are accurate.

L	d_l	epochs=1	epochs=5	epochs=10
2	32	11.96 (± 0.07)	11.97 (± 0.06)	11.98 (± 0.057)
2	128	11.96 (± 0.07)	11.97 (± 0.056)	11.97 (± 0.061)
2	512	11.95 (± 0.076)	11.95 (± 0.08)	11.96 (± 0.071)
4	32	11.93 (± 0.083)	11.94 (± 0.09)	11.96 (± 0.075)
4	128	11.89 (± 0.145)	11.93 (± 0.097)	11.95 (± 0.081)
4	512	11.86 (± 0.127)	11.93 (± 0.096)	11.94 (± 0.072)
8	32	11.89 (± 0.12)	11.93 (± 0.117)	11.95 (± 0.096)
8	128	11.88 (± 0.126)	11.92 (± 0.11)	11.94 (± 0.102)
8	512	11.85 (± 0.129)	11.9 (± 0.163)	11.92 (± 0.111)

Table 1: Put option with $r = 0.1$, $T = 1$, $K = 110$, $S_0 = 100$, $\sigma = 0.25$, $N = 10$ and $M = 100,000$. The true price is 11.987.

We can see from Table 1 that using a really small neural network with only one input layer with 32 intermediate neurons and one output layer — meaning that the activation function is applied only once — already yields very good results with a relative accuracy greater than 1%. Increasing the number of epochs helps correct the bias created by the truncated approximation of the conditional expectations. The larger the neural network (see in particular the cases $L = 8$), the more epochs we need to ensure that the fitting procedure has sufficiently well converged in order to make the most of the capabilities of the network to accurately approximate the conditional expectations. Note that increasing the size of the network also increases the overall variance of the algorithm as in the case of a polynomial regression when the size of the regression basis increases (see Glasserman and Yu [2004] for details).

5.1.2 A geometric basket option in the Black Scholes model

Benchmarking a new method on high dimensional products becomes hardly feasible as almost no high dimensional Bermudan options can be priced accurately in a reasonable time. An exception to this is the geometric put option with payoff $(K - (\prod_{j=1}^{d} S_t^j)^{1/d})_+$. Easy calculations show that the price of this $d-$dimensional option equals the one of the $1-$dimensional option with the following parameters

$$\hat{S}_0 = \left(\prod_{j=1}^{d} S_0^j\right)^{1/d}; \quad \hat{\sigma} = \frac{1}{d}\sqrt{\sigma^t \Gamma \sigma}; \quad \hat{\delta} = \frac{1}{d}\sum_{j=1}^{d}\left(\delta^j + \frac{1}{2}(\sigma^j)^2\right) - \frac{1}{2}(\hat{\sigma})^2.$$

In every numerical experiments on the geometric basket option, we report the price of the equivalent one dimensional Bermudan option obtained by the CRR tree method Cox et al. [1979] with $100,000$ discretization time steps.

L	d_l	epochs=1	epochs=5	epochs=10
2	32	4.55 (± 0.038)	4.56 (± 0.041)	4.56 (± 0.031)
2	128	4.55 (± 0.032)	4.56 (± 0.04)	4.56 (± 0.038)
2	512	4.54 (± 0.04)	4.55 (± 0.033)	4.55 (± 0.041)
4	32	4.52 (± 0.044)	4.54 (± 0.04)	4.55 (± 0.036)
4	128	4.52 (± 0.044)	4.54 (± 0.033)	4.55 (± 0.041)
4	512	4.5 (± 0.046)	4.54 (± 0.042)	4.54 (± 0.045)
8	32	4.52 (± 0.043)	4.54 (± 0.049)	4.55 (± 0.052)
8	128	4.51 (± 0.046)	4.53 (± 0.045)	4.54 (± 0.045)
8	512	4.47 (± 0.181)	4.51 (± 0.051)	4.52 (± 0.149)

Table 2: Prices for the geometric basket put option with parameters $d = 2$, $S_0^i = 100$, $\sigma^i = 0.2$, $\rho = 0$, $\delta^j = 0.2$, $T = 1$, $r = 0.05$, $K = 100$, $N = 10$ and $M = 100,000$. The true price is 4.57 (all digits are significant).

We can see from the numerical results of Table 2 that even a small neural network is able to capture the continuation values very well. Increasing the size of the network does not help get a better price but increases the variance unless we ensure a very accurate fit of the network by going through the data several times (see the column *epochs=10* for instance), which in turn leads to a much larger computational cost. In comparison, the prices obtained with the standard Longstaff Schwartz algorithm with polynomial regressions of order respectively 1, 3 and 6 are 4.47 ± 0.015, 4.56 ± 0.02 and 4.57 ± 0.017. On this small dimensional example, a low degree polynomial as well as a small neural network give a very accurate price.

L	d_l	epochs=1	epochs=5	epochs=10
2	32	2.91 (± 0.027)	2.92 (± 0.023)	2.93 (± 0.019)
2	128	2.91 (± 0.025)	2.93 (± 0.021)	2.94 (± 0.024)
2	512	2.9 (± 0.025)	2.93 (± 0.023)	2.94 (± 0.027)
4	32	2.9 (± 0.027)	2.92 (± 0.029)	2.94 (± 0.021)
4	128	2.9 (± 0.033)	2.92 (± 0.023)	2.93 (± 0.027)
4	512	2.89 (± 0.028)	2.91 (± 0.033)	2.93 (± 0.033)
8	32	2.9 (± 0.02)	2.92 (± 0.029)	2.94 (± 0.024)
8	128	2.9 (± 0.036)	2.92 (± 0.026)	2.94 (± 0.026)
8	512	2.88 (± 0.042)	2.91 (± 0.033)	2.92 (± 0.034)

Table 3: Prices for the geometric basket put option with parameters $d = 10$, $S_0^i = 100$, $\sigma^i = 0.2$, $\rho = 0.2$, $\delta^j = 0$, $T = 1$, $r = 0.05$, $K = 100$, $n = 10$ and $M = 100,000$. The true price is 2.92.

The numerical results for the $10-$dimensional geometric put option (see Table 3) show the same behavior as the low dimensional problem. Using a small neural network provides very accurate results within 0.1% of the true price. Passing several times over the data to train the network helps a little reduce the bias of the price estimator but at the expense of a much higher computational effort. In comparison, the prices obtained with the standard Longstaff Schwartz algorithm with polynomial regression of order respectively 1 and 3 are 2.86 ± 0.014 and 2.96 ± 0.014. Note that a regression of order 6 is unreachable in dimension 10. Unlike all other examples, in which the standard Longstaff Schwartz algorithm with polynomial regression tends to exhibit a systematic negative bias, increasing the polynomial degree in this example yields a price above the true one. Note that the true price is always within the confidence intervals reported in Table 3. Our method does not seem to suffer from this positive bias phenomenon.

L	d_l	epochs=1	epochs=5	epochs=10
2	128	2.52 (± 0.025)	2.57 (± 0.019)	2.61 (± 0.021)
2	256	2.51 (± 0.027)	2.57 (± 0.018)	2.61 (± 0.017)
2	512	2.5 (± 0.011)	2.56 (± 0.021)	2.61 (± 0.023)
4	128	2.51 (± 0.03)	2.59 (± 0.023)	2.78 (± 0.045)
4	256	2.51 (± 0.031)	2.57 (± 0.018)	2.75 (± 0.023)
4	512	2.49 (± 0.02)	2.55 (± 0.025)	2.65 (± 0.035)
8	128	2.51 (± 0.018)	2.58 (± 0.022)	2.76 (± 0.051)
8	256	2.51 (± 0.026)	2.57 (± 0.021)	2.75 (± 0.038)
8	512	2.46 (± 0.135)	2.56 (± 0.021)	2.65 (± 0.056)

Table 4: Prices for the geometric basket put option with parameters $d = 40$, $S_0^i = 100$, $\sigma^i = 0.2$, $\rho = 0.2$, $\delta^j = 0$, $T = 1$, $r = 0.5$, $K = 100$, $n = 10$ and $M = 100,000$. The true price is 2.52.

Finally, we tested our approach on a $40-$dimensional geometric basket option with two dif-

L	d_l	epochs=1	epochs=5	epochs=10
2	128	2.5 (± 0.008)	2.52 (± 0.005)	2.52 (± 0.005)
2	256	2.5 (± 0.012)	2.52 (± 0.005)	2.52 (± 0.01)
2	512	2.49 (± 0.015)	2.51 (± 0.007)	2.52 (± 0.009)
4	128	2.5 (± 0.006)	2.52 (± 0.006)	2.53 (± 0.003)
4	256	2.5 (± 0.009)	2.51 (± 0.007)	2.52 (± 0.005)
4	512	2.49 (± 0.008)	2.51 (± 0.011)	2.52 (± 0.014)
8	128	2.5 (± 0.007)	2.53 (± 0.011)	2.54 (± 0.008)
8	256	2.49 (± 0.012)	2.52 (± 0.011)	2.53 (± 0.007)
8	512	2.46 (± 0.154)	2.49 (± 0.053)	2.51 (± 0.015)

Table 5: Prices for the geometric basket put option with parameters $d = 40$, $S_0^i = 100$, $\sigma^i = 0.2$, $\rho = 0.2$, $\delta^j = 0$, $T = 1$, $r = 0.5$, $K = 100$, $n = 10$ and $M = 1,000,000$. The true price is 2.52

ferent number of Monte Carlo samples $M = 100,000$ (Table 4) and $M = 1,000,000$ (Table 5). In both cases, the results obtained with really small neural networks ($L = 2$) are already very accurate. However, one should note that increasing the number of epochs with $M = 100,000$ leads to upper biased prices. This is the result of an overfitting phenomenon during the neural network calibration. Remember that the number of parameters of the network is $(1+d)(1+d_l)+(d_l)^{2(L-2)}$. For $L = 4$ and $d_l = 128$, it already gives more than 26 million parameters. The learning capabilities of the network are such that it also learns the noise inside the data. Increasing the size of the data set ($M = 1,000,000$) fixes this issue as one can see in Table 5.

5.1.3 A put basket option

We consider a put basket option with payoff

$$\left(K - \sum_{i=1}^{d} \omega_i S_T^i\right)_+ .$$

We test our algorithm in dimension 5 and report the results in Table 6. The standard Longstaff Schwartz algorithm yields 3.11 ± 0.01 (resp. 3.05 ± 0.01) for an order 3 (resp. 1) polynomial regression. The prices reported in Table 6 are very close to the one obtained with an order 3 polynomial regression. We can see that using a very large neural networks with several hidden layers and several hundreds of neurons per layer does not really help. The results obtained for a small network with a few dozens of neurons are already very good. The difference between the results for 1 epoch and 10 epochs is about half the width of the confidence interval, which makes it non meaningful. Hence, there is no use putting more computational effort to go through all the data set more than once.

Now, we turn to a high-dimensional problem and consider a call option on a basket with 40 assets to test the scalability of a our approach. The results are reported in Table 7 for $M = 100,000$ Monte Carlo samples and in Table 8 for $M = 1,000,000$ Monte Carlo samples. As a

L	d_l	epochs=1	epochs=5	epochs=10
2	32	3.08 ($\pm$ 0.023)	3.09 ($\pm$ 0.023)	3.1 ($\pm$ 0.028)
2	128	3.08 ($\pm$ 0.024)	3.09 ($\pm$ 0.024)	3.1 ($\pm$ 0.027)
2	512	3.08 ($\pm$ 0.032)	3.09 ($\pm$ 0.023)	3.09 ($\pm$ 0.03)
4	32	3.07 ($\pm$ 0.032)	3.09 ($\pm$ 0.031)	3.1 ($\pm$ 0.027)
4	128	3.07 ($\pm$ 0.03)	3.09 ($\pm$ 0.027)	3.09 ($\pm$ 0.027)
4	512	3.06 ($\pm$ 0.038)	3.08 ($\pm$ 0.031)	3.09 ($\pm$ 0.03)
8	32	3.07 ($\pm$ 0.032)	3.09 ($\pm$ 0.028)	3.09 ($\pm$ 0.033)
8	128	3.06 ($\pm$ 0.035)	3.08 ($\pm$ 0.026)	3.1 ($\pm$ 0.027)
8	512	3.06 ($\pm$ 0.053)	3.07 ($\pm$ 0.053)	3.08 ($\pm$ 0.038)

Table 6: Basket option with $d = 5$, $r = 0.05$, $T = 1$, $\sigma^i = 0.2$, $\omega^i = 1/d$, $S_0^i = 100$, $\rho = 0.2$, $K = 100$, $N = 10$ and $M = 100,000$.

comparison, Goudenège et al. [2019] reported prices between 2.15 and 2.22 for the same option. Our prices lie in this interval. We note that increasing the number of epochs for a relatively small number of Monte Carlo samples $M = 100,000$ gives larger prices. This is all the more striking as the size of the neural network is large, which clearly exhibits an over-fitting phenomenon. Indeed, increasing the number of samples to $M = 1,000,000$ fixes the issue as the prices in Table 8 are between 2.14 and 2.18 for all the neural network configurations and the number of epochs up to 10. This clearly shows that to avoid an upper bias, we need to increase the number of samples when the size of the problem increases, which was already noted with the standard Longstaff Scwhartz algorithm using polynomial regression. With this example, we come to the same conclusions as for the $40-$dimensional geometric basket option studied in Section 5.1.2.

L	d_l	epochs=1	epochs=5	epochs=10
2	32	2.15 ($\pm$ 0.018)	2.19 ($\pm$ 0.019)	2.21 ($\pm$ 0.02)
2	128	2.16 ($\pm$ 0.016)	2.21 ($\pm$ 0.015)	2.25 ($\pm$ 0.021)
2	512	2.15 ($\pm$ 0.017)	2.21 ($\pm$ 0.014)	2.26 ($\pm$ 0.017)
4	32	2.16 ($\pm$ 0.018)	2.21 ($\pm$ 0.015)	2.26 ($\pm$ 0.017)
4	128	2.16 ($\pm$ 0.021)	2.24 ($\pm$ 0.024)	2.43 ($\pm$ 0.026)
4	512	2.15 ($\pm$ 0.018)	2.2 ($\pm$ 0.025)	2.31 ($\pm$ 0.026)
8	32	2.17 ($\pm$ 0.028)	2.21 ($\pm$ 0.02)	2.28 ($\pm$ 0.023)
8	128	2.16 ($\pm$ 0.026)	2.24 ($\pm$ 0.025)	2.41 ($\pm$ 0.032)
8	512	2.14 ($\pm$ 0.064)	2.19 ($\pm$ 0.031)	2.29 ($\pm$ 0.044)

Table 7: Basket option with $d = 40$, $r = 0.05$, $T = 1$, $\sigma^i = 0.2$, $\omega^i = 1/d$, $S_0^i = 100$, $\rho = 0.2$, $K = 100$, $N = 10$ and $M = 100,000$.

L	d_l	epochs=1	epochs=5	epochs=10
2	32	2.16 (± 0.008)	2.17 (± 0.008)	2.18 (± 0.009)
2	128	2.16 (± 0.009)	2.17 (± 0.008)	2.17 (± 0.007)
2	512	2.15 (± 0.01)	2.17 (± 0.007)	2.17 (± 0.005)
4	32	2.17 (± 0.008)	2.17 (± 0.008)	2.18 (± 0.007)
4	128	2.16 (± 0.012)	2.17 (± 0.008)	2.18 (± 0.007)
4	512	2.15 (± 0.014)	2.16 (± 0.01)	2.16 (± 0.01)
8	32	2.16 (± 0.011)	2.18 (± 0.009)	2.18 (± 0.006)
8	128	2.16 (± 0.015)	2.17 (± 0.007)	2.18 (± 0.007)
8	512	2.14 (± 0.022)	2.15 (± 0.056)	2.16 (± 0.015)

Table 8: Basket option with $d = 40$, $r = 0.05$, $T = 1$, $\sigma^i = 0.2$, $\omega^i = 1/d$, $S_0^i = 100$, $\rho = 0.2$, $K = 100$, $N = 10$ and $M = 1,000,000$.

5.1.4 A call on the maximum of several assets in the Black Scholes model

We consider a call option on the maximum of d assets in the Black Scholes model with payoff

$$\left(\max_{i=1,\dots,d} S_T^i - K\right)_+ .$$

The different sets of parameters are chosen as in Becker et al. [2019a], Goudenège et al. [2019] to easily compare the prices obtained with the different methods.

L	d_l	epochs=1	epochs=5	epochs=10
2	32	25.97 (± 0.117)	25.95 (± 0.141)	25.94 (± 0.133)
2	128	25.95 (± 0.11)	25.95 (± 0.126)	26.02 (± 0.113)
2	512	25.92 (± 0.104)	25.96 (± 0.116)	26.01 (± 0.153)
4	32	25.83 (± 0.132)	25.97 (± 0.146)	26.02 (± 0.139)
4	128	25.76 (± 0.203)	25.91 (± 0.162)	25.99 (± 0.162)
4	512	25.63 (± 0.238)	25.85 (± 0.181)	25.94 (± 0.146)
8	32	25.72 (± 0.185)	25.91 (± 0.134)	25.96 (± 0.169)
8	128	25.61 (± 0.251)	25.84 (± 0.186)	25.93 (± 0.143)
8	512	25.49 (± 0.265)	25.76 (± 0.223)	25.83 (± 0.2)

Table 9: Prices for the call option on the maximum of 5 assets with parameters $S_0^j = 100$, $T = 3$, $r = 0.05$, $K = 100$, $\rho = 0$, $\sigma^j = 0.2$, $\delta^j = 0.1$, $N = 9$ and $M = 100,000$.

Pricing a call on the maximum of a basket of assets is usually far more difficult than a standard basket option because of the strong non linearity of the maximum function. For the example of Table 9, the standard Longstaff Schwartz algorithm yields 25.34 ± 0.06, 25.98 ± 0.05, 26.29 ± 0.04 for a polynomial regression of order 1, 3 and 6 respectively. The prices obtained with the polynomial regression vary a lot with the degree of the regression. For this example, Becker et al. [2019a] reported a 95% confidence interval of $[26.14, 26.17]$. The prices reported in Table 9 are very close to this confidence interval. A small neural network ($L = 2$) enables us to get values within 1% of the true price, which is a great achievement considering the complexity of the product and the small size of the approximation. As in the other examples, using several passes through the data to train the neural network does not really bring any improvement for small neural networks. For larger networks, it helps a little but in the end larger networks are less accurate than smaller ones. To get the best of larger neural networks, we would need more data to train the networks, ie. more Monte Carlo samples as we already observed for the high dimensional geometric put option.

L	d_l	epochs=1	epochs=5	epochs=10
2	128	68.99 ($\pm$ 0.179)	69.26 ($\pm$ 0.164)	69.42 ($\pm$ 0.169)
2	256	69.07 ($\pm$ 0.149)	69.42 ($\pm$ 0.125)	69.45 ($\pm$ 0.138)
2	512	69.11 ($\pm$ 0.194)	69.43 ($\pm$ 0.18)	69.51 ($\pm$ 0.167)
4	128	68.91 ($\pm$ 0.365)	69.29 ($\pm$ 0.334)	69.55 ($\pm$ 0.339)
4	256	68.72 ($\pm$ 0.358)	69.24 ($\pm$ 0.341)	69.5 ($\pm$ 0.369)
4	512	68.54 ($\pm$ 0.548)	69.17 ($\pm$ 0.356)	69.34 ($\pm$ 0.359)
8	128	68.59 ($\pm$ 0.613)	69.32 ($\pm$ 0.348)	69.71 ($\pm$ 0.497)
8	256	68.57 ($\pm$ 0.797)	69.25 ($\pm$ 0.564)	69.4 ($\pm$ 0.484)
8	512	68.32 ($\pm$ 1.444)	69.01 ($\pm$ 0.738)	69.49 ($\pm$ 0.487)

Table 10: Prices for the call option on the maximum of 50 assets with parameters $S_0^j = 100$, $T = 3$, $r = 0.05$, $K = 100$, $\rho = 0$, $\sigma^j = 0.2$, $\delta^j = 0.1$, $N = 9$ and $M = 100,000$.

L	d_l	epochs=1	epochs=5	epochs=10
2	128	68.85 ($\pm$ 0.074)	68.96 ($\pm$ 0.095)	69.01 ($\pm$ 0.119)
2	256	68.87 ($\pm$ 0.1)	69.0 ($\pm$ 0.143)	69.07 ($\pm$ 0.146)
2	512	68.82 ($\pm$ 0.082)	69.05 ($\pm$ 0.128)	69.19 ($\pm$ 0.136)
4	128	68.84 ($\pm$ 0.221)	69.28 ($\pm$ 0.153)	69.41 ($\pm$ 0.211)
4	256	68.75 ($\pm$ 0.342)	69.14 ($\pm$ 0.296)	69.38 ($\pm$ 0.342)
4	512	68.7 ($\pm$ 0.426)	69.05 ($\pm$ 0.317)	69.35 ($\pm$ 0.254)
8	128	68.81 ($\pm$ 0.277)	69.28 ($\pm$ 0.291)	69.64 ($\pm$ 0.22)
8	256	68.57 ($\pm$ 0.512)	69.34 ($\pm$ 0.378)	69.65 ($\pm$ 0.414)

Table 11: Prices for the call option on the maximum of 50 assets with parameters $S_0^j = 100$, $T = 3$, $r = 0.05$, $K = 100$, $\rho = 0$, $\sigma^j = 0.2$, $\delta^j = 0.1$, $N = 9$ and $M = 1,000,000$.

We tested the scalability of our approach on a $50-$dimensional max-call option. The results are reported in Table 10 for $M = 100,000$ and Table 11 for $M = 1,000,000$. Becker et al. [2019a] report $[69.56, 69.95]$ as the 95% confidence interval for the option price. We obtain prices very close these values. As a comparison, the standard Longstaff Schwartz algorithm yields 67.96 ± 0.02 and 69.02 ± 0.02 for a polynomial regression of order 1 and 2 respectively. The order 3 regression is out of reach. Our deep learning approach scales far better than the standard polynomial regression. Increasing the number of epochs improves the result, which means that the neural network has to be much more finely tuned than in the other examples studied sofar.

5.2 A put option in the Heston model

We consider the Heston model defined by

$$\begin{aligned} dS_t &= S_t(r_t dt + \sqrt{\sigma_t}(\rho dW_t^1 + \sqrt{1-\rho^2} dW_t^2)) \\ d\sigma_t &= \kappa(\theta - \sigma_t)dt + \xi\sqrt{\sigma_t} dW_t^1. \end{aligned}$$

For the simulation of the model, we use a modified Euler scheme with 30 time steps per year, in which we have replaced $\sqrt{\sigma_t}$ by $\sqrt{(\sigma_t)_+}$ to deal with possibly negative values of the discretized volatility process. For the option of Table 12, the standard Longstaff Schwartz algorithm yields 1.70 ± 0.008 (resp. 1.675 ± 0.005) for an order 6 (resp. 1) polynomial regression. As in the other examples, the use of a neural network as the regressor provides very accurate results even with a quite small network (no hidden layer and very few neurons, see the case $L = 2$ and $d_l = 32$).

L	d_l	epochs=1	epochs=5	epochs=10
2	32	1.69 (± 0.017)	1.7 (± 0.017)	1.7 (± 0.016)
2	128	1.69 (± 0.017)	1.7 (± 0.019)	1.7 (± 0.019)
2	512	1.69 (± 0.019)	1.69 (± 0.019)	1.69 (± 0.018)
4	32	1.69 (± 0.022)	1.69 (± 0.017)	1.7 (± 0.018)
4	128	1.69 (± 0.024)	1.69 (± 0.02)	1.7 (± 0.016)
4	512	1.68 (± 0.025)	1.69 (± 0.022)	1.69 (± 0.022)
8	32	1.69 (± 0.023)	1.69 (± 0.02)	1.69 (± 0.019)
8	128	1.68 (± 0.03)	1.69 (± 0.022)	1.69 (± 0.02)
8	512	1.68 (± 0.03)	1.68 (± 0.041)	1.68 (± 0.053)

Table 12: Prices for put option in the Heston model with parameters the geometric basket put option with parameters with $S_0 = K = 100$, $T = 1$, $\sigma_0 = 0.01$, $\xi = 0.2$, $\theta = 0.01$, $\kappa = 2$, $\rho = -0.3$, $r = 0.1$, $N = 10$ and $M = 100,000$.

6 Conclusion

The difficulties in pricing Bermudan options come from approximating the continuation value at each exercising date. While polynomial regression is widely used for this step, we have investigated the use of deep learning. We have proved the theoretical convergence of our algorithm with respect to both the neural network and Monte Carlo approximations. Our numerical experiments show that the prices computed using our approach are very similar to those obtained from the standard Longstaff Schwartz algorithm. With no surprise, using neural networks does not help much for low dimensional problems but does scale far better on high dimensional problems as it does not suffer from the curse of dimensionality as much as polynomial regression does. Polynomial regression requires a relatively high order to provide accurate prices, which is not feasible in high dimensional problems. Neural networks approximation capabilities seem far better and relatively small networks already provided very accurate results. Indeed, a few hundred neurons with no hidden layers were sufficient to have very accurate prices. Training a neural network usually requires several passes through the whole data set. Yet, in our examples this seemed pretty much useless mostly because the functional representation of the continuation function should not vary much over time. So, once the neural network has been well trained, one pass over the data ($epochs = 1$) is enough to fit the network at a new date. This saves a lot of computational time. Neural networks have proved to be a very versatile and efficient tool to compute Bermudan option prices especially when the problem is highly non linear.

References

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

F. Albertini and E. D. Sontag. Uniqueness of weights for recurrent nets. *MATHEMATICAL RESEARCH*, 79:599–599, 1994.

F. Albertini, E. D. Sontag, and V. Maillot. Uniqueness of weights for neural networks. *Artificial Neural Networks for Speech and Vision*, pages 115–125, 1993.

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

V. Bally and G. Pages. A quantization algorithm for solving multidimensional discrete-time optimal stopping problems. *Bernoulli*, 9(6):1003–1049, 2003.

S. Becker, P. Cheridito, and A. Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20(74):1–25, 2019a.

S. Becker, P. Cheridito, A. Jentzen, and T. Welti. Solving high-dimensional optimal stopping problems using deep learning, 2019b.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

B. Bouchard and X. Warin. Monte-carlo valuation of american options: Facts and new algorithms to improve existing methods. In R. A. Carmona, P. Del Moral, P. Hu, and N. Oudjane, editors, *Numerical Methods in Finance*, volume 12 of *Springer Proceedings in Mathematics*, pages 215–255. Springer Berlin Heidelberg, 2012.

M. Broadie and P. Glasserman. A stochastic mesh method for pricing high-dimensional american options. *Journal of Computational Finance*, 7:35–72, 2004.

A. L. Bronstein, G. Pagès, and J. Portès. Multi-asset american options and parallel quantization. *Methodology and Computing in Applied Probability*, 15(3):547–561, 2013.

J. F. Carriere. Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: mathematics and Economics*, 19(1):19–30, 1996.

E. Clément, D. Lamberton, and P. Protter. An analysis of a least squares regression method for american option pricing. *Finance and Stochastics*, 6(4):449–471, 2002.

J. Cox, S. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, (7):229–263, 1979.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.

W. E, C. Ma, S. Wojtowytsch, and L. Wu. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't, 2020.

F. Fang and C. W. Oosterlee. Pricing early-exercise and discrete barrier options by fourier-cosine series expansions. *Numerische Mathematik*, 114(1):27, 2009.

B. Fehrman, B. Gess, and A. Jentzen. Convergence rates for the stochastic gradient descent method for non-convex objective functions. *Journal of Machine Learning Research*, 21(136): 1–48, 2020.

S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

P. Glasserman and B. Yu. Number of paths versus number of basis functions in american option pricing. *The Annals of Applied Probability*, 14(4):2090–2119, 2004.

E. Gobet, J. Lemor, and X. Warin. A regression-based Monte Carlo method to solve backward stochastic differential equations. *Annals of Applied Probability*, 15(3):2172–2202, 2005.

L. Goudenège, A. Molent, and A. Zanette. Variance reduction applied to machine learning for pricing bermudan/american options in high dimension. *arXiv preprint arXiv:1903.11275*, 2019.

M. B. Haugh and L. Kogan. Pricing american options: a duality approach. *Operations Research*, 52(2):258–270, 2004.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4 (2):251 – 257, 1991.

M. Kohler, A. Krzyżak, and N. Todorovic. Pricing of high-dimensional american options by neural networks. *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics*, 20(3):383–410, 2010.

A. Kolmogorov. On the representation of continuous functions of several variables as superpositions of functions of smaller number of variables. In *Soviet. Math. Dokl*, volume 108, pages 179–182, 1956.

M. Ledoux and M. Talagrand. *Probability in Banach spaces*, volume 23 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)].* Springer-Verlag, Berlin, 1991. ISBN 3-540-52013-9. Isoperimetry and processes.

Y. Lei, T. Hu, G. Li, and K. Tang. Stochastic gradient descent for nonconvex learning without bounded gradient assumptions. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

F. Longstaff and R. Schwartz. Valuing American options by simulation : A simple least-square approach. *Review of Financial Studies*, 14:113–147, 2001.

R. Lord, F. Fang, F. Bervoets, and C. W. Oosterlee. A fast and accurate fft-based method for pricing early-exercise options under lévy processes. *SIAM Journal on Scientific Computing*, 30(4):1678–1705, 2008.

G. Pagès. *Numerical Probability: An Introduction with Applications to Finance*. Springer, 2018. doi: 10.1007/978-3-319-90276-0.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

A. Pinkus. Approximation theory of the mlp model in neural networks. *Acta numerica*, 8: 143–195, 1999.

R. Y. Rubinstein and A. Shapiro. *Discrete event systems*. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons Ltd., Chichester, 1993. ISBN 0-471-93419-4. Sensitivity analysis and stochastic optimization by the score function method.

J. A. Tilley. Valuing american options in a path simulation model. *Transactions of the Society of Actuaries*, 45(83):104, 1993.

J. Tsitsiklis and B. V. Roy. Regression methods for pricing complex American-style options. *IEEE Trans. Neural Netw.*, 12(4):694–703, 2001.

R. C. Williamson and U. Helmke. Existence and uniqueness results for neural network approximations. *IEEE Transactions on Neural Networks*, 6(1):2–13, 1995.