Journal of Machine Learning Research 24 (2023) 1-51 Submitted 8/21; Revised 12/22; Published 7/23

Random Feature Neural Networks Learn Black-Scholes Type PDEs Without Curse of Dimensionality

Lukas Gonon L.GONON@IMPERIAL.AC.UK
Department of Mathematics
Imperial College London
UK

Editor: Ingo Steinwart

Abstract

This article investigates the use of random feature neural networks for learning Kolmogorov partial (integro-)differential equations associated to Black-Scholes and more general exponential Lévy models. Random feature neural networks are single-hidden-layer feedforward neural networks in which the hidden weights are randomly generated and only the output weights are trainable. This makes training particularly simple, but (a priori) reduces expressivity. Interestingly, this is not the case for certain Black-Scholes type PDEs, as we show here. We derive bounds for the prediction error of random neural networks for learning sufficiently non-degenerate Black-Scholes type models. A full error analysis – bounding the approximation, generalization and optimization error of the algorithm – is provided and it is shown that the derived bounds do not suffer from the curse of dimensionality. We also investigate an application of these results to basket options and validate the bounds numerically.

These results prove that neural networks are able to *learn* solutions to suitable Black-Scholes type PDEs without the curse of dimensionality. In addition, this provides an example of a relevant learning problem in which random feature neural networks are provably efficient.

Keywords: Random features, neural networks, Black-Scholes model, exponential Lévy model, generalization bounds, curse of dimensionality

1. Introduction

A fundamental problem in science and engineering is to infer an unknown input-output relation from data. In recent years (artificial) neural networks have become an important tool to address such problems in complex, high-dimensional situations. Neural networks have shown a strikingly efficient computational performance in an enormous range of applications and impressive progress has also been made regarding the theoretical and mathematical foundations of neural network-based methods.

In many situations additional a priori information about the unknown input-output relation is available and the problem amounts to learning the solution of a partial differential equation (PDE) or, for instance in a financial context, an expectation of a stochastic process. Examples of applications of neural networks in this area can be found e.g. in Han et al. (2018), E et al. (2017), Sirignano and Spiliopoulos (2018), Huré et al. (2020), Buehler et al. (2019), Cuchiero et al. (2020). We refer to the surveys Ruf and Wang (2020),

Beck et al. (2023), Germain et al. (2021) for an overview of the numerous recent applications of neural network-based learning in the context of PDEs, stochastic processes and finance. There has also been important progress regarding the theoretical and mathematical foundations of neural network-based methods in this area, see again the surveys mentioned above for an overview. Many of these recent mathematical results prove that deep neural networks are able to approximate solutions to various classes of Kolmogorov PDEs without the curse of dimensionality, see, for instance, Elbrächter et al. (2022), Grohs et al. (2023), Hutzenthaler et al. (2020), Reisinger and Zhang (2020), Laakmann and Petersen (2021), Kutyniok et al. (2022), Gonon and Schwab (2023). In some articles then a learning problem is considered and such approximation error bounds are combined with generalization error bounds in order to prove that the empirical risk-minimizing deep neural network is capable of overcoming the curse of dimensionality for learning solutions to certain PDEs, see, e.g., Berner et al. (2020), Carmona and Laurière (2021). In practice, the neural network that minimizes the empirical risk needs to be calculated approximately, which is typically achieved using a variant of the stochastic gradient descent algorithm. This introduces a further error component, the optimization error, which has remained challenging to analyze mathematically for general neural networks. As a consequence, in the context of Kolmogorov PDEs there have been no results in the literature so far which address all three error components and explain mathematically the success of neural networks at *learning* solutions to high-dimensional Kolmogorov PDEs. In this work such an explanation is provided by proving that neural networks are capable of learning solutions to certain Kolmogorov PDEs without the curse of dimensionality. This is achieved by considering neural networks in which only certain weights are trainable and the remaining parameters are generated randomly, as we will describe in more detail below. We mention that for other classes of PDEs mathematical results that provide an error analysis of learning by neural networks have been obtained, for instance, in Luo and Yang (2020) and Lu et al. (2021), where the authors consider certain classes of second-order PDEs on bounded domains. In this work we are concerned with Kolmogorov PDEs associated to stochastic processes naturally defined on unbounded domains.

We investigate the capabilities of random (feature) neural networks – see Huang et al. (2006), Rahimi and Recht (2008), Rahimi and Recht (2009) – as a learning method in the context of certain Kolmogorov PDEs. Random neural networks are feedforward neural networks with a single hidden layer and the property that the parameters of the hidden layer are randomly initialized and then fixed. Hence, only the parameters of the output layer can be trained. The non-convex optimization problem that needs to be solved in order to train a standard neural network reduces to a convex optimization problem here. This simplifies both training in practice and theoretical analysis. On the other hand, allowing only parts of the parameters to be trained reduces the approximation capabilities and so, at least a priori, it is not clear if random neural networks still have any of the powerful approximation properties of general deep neural networks. In several other contexts these questions have been addressed and learning (or prediction/test) error bounds for random features or random neural networks have been proved (see for instance Rahimi and Recht (2008), Rudi and Rosasco (2017), Carratino et al. (2018), Mei and Montanari (2022), Mei et al. (2022) and the references therein), but not in the context of PDEs. This is precisely the subject of this article. We investigate these questions for the problem of learning an

unknown function from a class of Kolmogorov PDEs, which include the Black-Scholes PDE as a special case. These partial (integro-)differential equations, which are also referred to as *(non-local) PDEs*, arise for instance in the context of option pricing in exponential Lévy models, see, e.g., Cont and Tankov (2004), Eberlein and Kallsen (2019) and the references therein.

The main results of this article prove that, indeed, random neural networks are capable of learning certain non-degenerate Black-Scholes type PDEs without the curse of dimensionality. We provide a full error analysis, i.e., bounds on the approximation error, the generalization (or estimation) error and the optimization error. For each of these error components we obtain polynomial convergence rates which do not depend on the dimension d of the underlying PDE and constants which grow at most polynomially in d. We refer to Corollary 27 for the precise statement. Corollary 27 is the main result of the article. It builds on the approximation error bound obtained in Theorem 13 and the learning error bounds for convolutional functions with fixed input dimension d obtained in Theorem 17, Theorem 19 and Proposition 21 for training by regression, constrained regression and stochastic gradient descent, respectively. Our proofs of these results employ techniques from statistical learning theory (in particular Rademacher complexities), results from Györfi et al. (2002), Shamir and Zhang (2013) and an L^∞-error bound for random neural network approximations provided in Theorem 1 and partially generalizing the L^2-error bound from Gonon et al. (2023a, Theorem 1).

The article contributes to the literature in several aspects. Firstly, it provides a fully implementable neural network-based algorithm for learning suitable Kolmogorov PDEs for which a full error analysis (covering all three error components) is available and which does not suffer from the curse of dimensionality. The solution to the PDE can be learnt on a full hypercube from observational data even without knowing the parameters of the PDE. Secondly, it provides an example of a practically relevant learning problem in which random features are provably efficient. Finally, the techniques developed in the article may also be helpful for the theoretical analysis of more general neural network-based learning methods in future works.

From the perspective of financial applications, the implication of the results is as follows. Consider the problem of learning prices of options from market data. For example, market prices of call options corresponding to certain strikes are observed and the aim is to predict call option prices for other strikes based on this data. The results obtained in this paper imply that in certain situations this learning problem can be solved efficiently using random neural networks, provided that there exists some exponential Lévy model (that is unknown and whose parameters do not need to be estimated) which calibrates well to observed market prices.

Neural networks with randomly sampled weights already appear in Barron's work (Barron, 1992, Barron, 1993). The random sampling-based dimension-independent convergence rates obtained there were also extended to the larger class of *generalized Barron functions* in E et al. (2020), E et al. (2019), E and Wojtowytsch (2020), see also Berner et al. (2021, Section 4.2) and Gonon et al. (2023b). For further related results and extensions we refer, for instance, to Barron and Klusowski (2018), Siegel and Xu (2020), Caragea et al. (2023) and the references therein. In all these results, the random sampling procedure is an intermediate step to establish the existence of neural network weights and obtain approximation

bounds. This does not yield a constructive sampling procedure in general, since the random sampling distribution depends on the unknown target function. In contrast, in the random features approach Rahimi and Recht (2009), Rahimi and Recht (2008) considered here the distribution from which the random weights are sampled is chosen a priori and does not depend on the target function. Our results build on a random feature neural network approximation error bound that applies to a subset of all *Barron functions* (as considered in Barron, 1992, Barron, 1993), see Theorem 1 below. The non-degeneracy condition that we impose on the considered Black-Scholes type models ensures that the solution is sufficiently smooth in the sense that its Fourier transform satisfies a certain integrability condition. For such functions the existence of an approximating (deterministic) shallow neural network is also asserted by Barron (1993), but a (high-dimensional) learning error analysis building on this result would require dimension-sensitive bounds on the optimization error (e.g. when employing the stochastic gradient descent algorithm to learn the hidden weights) in this context, which are presently not available. For a more detailed discussion of the relation between (deterministic) fully-trainable neural networks and random feature neural networks in the present context we refer to Section 5.3 below.

Numerical methods for partial (integro-)differential equations associated to univariate and certain multivariate exponential Lévy models were developed, e.g., in Cont and Voltchkova (2005), Farkas et al. (2007), Matache et al. (2004), Hilber et al. (2009). In a high-dimensional setting, when the parameters of the PDE are known and the solution of the PDE needs to be evaluated at a single point, then Monte Carlo methods are able to approximate the solution of the PDE without the curse of dimensionality. In contrast, here we consider a more challenging situation, which includes both the problem of evaluating the solution of the PDE on a full hypercube $[-M, M]^d$ by a numerical method and the problem of learning the solution of the PDE from observed values. In the latter situation, in particular, the true parameters of the PDE are unknown.

The remainder of the article is structured as follows. Section 2 introduces random neural networks and provides a general random neural network approximation error bound. In Section 3 we build on this result to provide random neural network approximation bounds for a class of convolutional functions and then specialize to the case of partial (integro-)differential equations or (non-local) PDEs associated to exponential Lévy models. Section 4 introduces the learning problem, provides error bounds for different learning methods (regression, constrained regression and stochastic gradient descent) and develops an application to basket option pricing. These results are then applied in Section 5 to prove that random neural networks are capable of learning certain Black-Scholes type PDEs without the curse of dimensionality. The paper concludes with numerical experiments to validate the obtained bounds.

1.1 Notation

In most parts of the article we will consider the dimension $d \in \mathbb{N}$ as fixed, but we will work out explicitly the dependence of all constants on d. In Sections 3.2 and 5 we will consider a family of models indexed by $d \in \mathbb{N}$ and thus d appears explicitly in the notation there.

Throughout, $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^d$ or $\mathbb{R}^N$ (the appropriate space will always be clear from the context). For $M > 0$ we denote the Euclidean ball by $B_M(0) =$

$\{x \in \mathbb{R}^d \,|\, \|x\| \leq M\}$. All random variables are defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and we write $\|\cdot\|_{L^\infty(\mathbb{P})} = \|\cdot\|_{L^\infty(\Omega,\mathcal{F},\mathbb{P})}$ for the L^∞-norm on $(\Omega, \mathcal{F}, \mathbb{P})$. For $x \in \mathbb{R}^d$ we use the notation $\exp(x) = (\exp(x_1), \ldots, \exp(x_d))$.

2. Random Neural Networks: Preliminary Results

In this section we recall the definition of random (feature) neural networks and provide a general approximation result. Such networks will be used to learn an unknown target function.

A random neural network is a feedforward neural network with one hidden layer and randomly generated hidden weights. More specifically, let $N \in \mathbb{N}$, let $B_1, \ldots, B_N$ be i.i.d. random variables, let $A_1, \ldots, A_N$ be i.i.d. $\mathbb{R}^d$-valued random vectors, assume that $A = (A_1, \ldots, A_N)$ and $B = (B_1, \ldots, B_N)$ are independent and for an $\mathbb{R}^N$-valued random vector W consider the (random) function

$$H_W^{A,B}(x) := \sum_{i=1}^{N} W_i \varrho(A_i \cdot x + B_i), \quad x \in \mathbb{R}^d, \tag{1}$$

where $\varrho\colon \mathbb{R} \to \mathbb{R}$ is a fixed activation function. Throughout the article we will consider random neural networks with the ReLU activation function given by $\varrho(z) = \max(z, 0)$ for $z \in \mathbb{R}$. The random variables A and B will be referred to as the (random) hidden weights of the neural network and W as the vector of output weights.

To approximate an unknown function $H\colon \mathbb{R}^d \to \mathbb{R}$ the (random) hidden weights A, B are considered as fixed and only the output vector W can be trained. Thus, the goal is to find W such that the expected uniform approximation error $\mathbb{E}[\|H_W^{A,B} - H\|_{L^\infty([-M,M]^d)}]$ is small.

Approximation properties of such random neural networks have been studied for instance in Huang et al. (2006), Rahimi and Recht (2008), Rudi and Rosasco (2017) and most recently in Gonon et al. (2023a). Theorem 1 below is a novel approximation result for sufficiently regular functions, which will be crucial for the results in Section 3. The result and parts of the proof of Theorem 1 are similar to Gonon et al. (2023a, Theorem 1); however in Gonon et al. (2023a, Theorem 1) a more general Hilbert space setting and more general sampling distributions are considered. In contrast, Theorem 1 works under stronger hypotheses and employs Rademacher complexity-based techniques to obtain a *uniform error bound* instead of an L^2-error bound.

More specifically, in what follows we make the following assumptions on the distribution of the hidden weights of the random neural network (1):

- the distribution of A_1 has a strictly positive Lebesgue-density π_{w} on $\mathbb{R}^d$ and
- the distribution of B_1 has a strictly positive Lebesgue-density π_{b} on $\mathbb{R}$.

In this situation, the following random neural network approximation result holds.

Theorem 1 *Let $H\colon \mathbb{R}^d \to \mathbb{R}$, let $M > 0$ and assume there exists $G\colon \mathbb{R}^d \to \mathbb{C}$ such that*

$$H(x) = \int_{\mathbb{R}^d} e^{ix\cdot\xi} G(\xi) d\xi \tag{2}$$

for all $x \in [-M,M]^d$. *Suppose that*

$$\int_{\mathbb{R}^d} \max(1, \|\xi\|^2)|G(\xi)|d\xi < \infty, \tag{3}$$

$\bar{F}(r) := 2\int_{-r}^{0} \frac{1}{\pi_b(s)} ds \in (-\infty, \infty)$ *for all* $r \in \mathbb{R}$ *and*

$$I = \max(16, M^2) \int_{\mathbb{R}^d} [\bar{F}(M\|\xi\|_1)\|\xi\|_1^2 + (\bar{F}(1) - \bar{F}(-1)) \max(1, \|\xi\|^2)] \frac{(|G(\xi)| + |G(-\xi)|)^2}{\pi_w(\xi)} d\xi \tag{4}$$

is finite. Then there exists an $\mathbb{R}^N$*-valued,* $\sigma(A,B)$*-measurable random vector* W *such that*

$$\mathbb{E}\left[\sup_{x\in[-M,M]^d} |H_W^{A,B}(x) - H(x)|\right] \leq \frac{4(M\sqrt{d}+1)\sqrt{I}}{\sqrt{N}}. \tag{5}$$

Moreover, $\|W_i\|_{L^\infty(\mathbb{P})} \leq \frac{1}{N} \sup_{(u,\xi)\in\mathbb{R}\times\mathbb{R}^d} (\mathbb{1}_{[-M\|\xi\|_1,0]}(u) + 4\mathbb{1}_{[-1,1]}(u)) \frac{|G(\xi)|+|G(-\xi)|}{\pi_b(u)\pi_w(\xi)}$ *for* $i = 1, \ldots, N$.

Remark 2 *The proof of Theorem 1 is based on several ingredients: firstly,* (2) *is used to derive an integral representation for* H *(see* (10)*). This representation is related to the Radon-wavelet integral representation (as used in Maiorov and Meir, 2000) and representations in Barron (1992), Barron (1993), Klusowski and Barron (2018). Secondly, the output weights* W *are selected based on an "importance sampling procedure" (see* (11)*). This matches the distribution of the random weights (which is chosen a priori and does not depend on* H*) with the function* α *in the integral representation for* H *(see* (10)*). Thirdly, Rademacher complexity-based techniques (Bartlett and Mendelson 2003, Boucheron et al. 2013, Ledoux and Talagrand 2013) are employed to bound the* L^∞*-error between the random neural network and the target function* H *on the hypercube* $[-M,M]^d$*. The first two ingredients were also used in the proof of Gonon et al. (2023a, Theorem 1).*

Proof First, let us point out that for any $\mathbb{R}^N$-valued random vector W the mapping $(\omega, x) \mapsto H_{W(\omega)}^{A(\omega),B(\omega)}(x) = \sum_{i=1}^N W_i(\omega)\varrho(A_i(\omega)\cdot x + B_i(\omega))$ is $\mathcal{F}\otimes\mathcal{B}(\mathbb{R}^d)$-measurable by Aliprantis and Border (2006, Lemma 4.51).

We now proceed in two steps. The first step consists in deriving an integral representation of H based on (2), as in Gonon et al. (2023a). From this integral representation we construct the output weights W based on an importance sampling procedure. The second step then uses Rademacher complexities to estimate the expected L^∞-error.

Step 1: By considering separately the cases $r > 0$ and $r < 0$, one obtains for any $r \in \mathbb{R}$ the identity

$$e^{ir} - ir - 1 = -\int_0^\infty (r-u)^+ e^{iu} + (-r-u)^+ e^{-iu} du. \tag{6}$$

Inserting $r = \xi\cdot x$, multiplying by $G(\xi)$, integrating over $\xi \in \mathbb{R}^d$, employing the representation (2) and using Fubini's theorem (which can be applied due to (3)) hence yields for any $x \in [-M,M]^d$ that

$$\begin{aligned} H(x) - x\cdot\nabla H(0) - H(0) &= \int_{\mathbb{R}^d} e^{ix\cdot\xi} G(\xi) - ix\cdot\xi G(\xi) - G(\xi) d\xi \\ &= -\int_0^\infty \int_{\mathbb{R}^d} [(x\cdot\xi - u)^+ e^{iu} + (-x\cdot\xi - u)^+ e^{-iu}] G(\xi) d\xi du. \end{aligned} \tag{7}$$

Changing variables in the integral and using that for $x \in [-M, M]^d$ and $u \leq -M\|\xi\|_1$ we have $(x \cdot \xi + u)^+ = 0$ then shows for all $x \in [-M, M]^d$ that

$$H(x) - x \cdot \nabla H(0) - H(0) = -\int_{\mathbb{R}^d} \int_{-M\|\xi\|_1}^{0} (x \cdot \xi + u)^+ [e^{-iu} G(\xi) + e^{iu} G(-\xi)] du d\xi. \tag{8}$$

From the fact that $H(0)$ and $\nabla H(0)$ are elements of $\mathbb{R}$ one obtains $\int_{\mathbb{R}^d} \mathrm{Im}[G](\xi) d\xi = 0$ and $\int_{\mathbb{R}^d} \xi \mathrm{Re}[G](\xi) d\xi = 0$ and hence we can represent

$$\begin{aligned} x \cdot \nabla H(0) + H(0) &= \int_{\mathbb{R}^d} -(x \cdot \xi) \mathrm{Im}[G](\xi) d\xi + \int_{\mathbb{R}^d} \mathrm{Re}[G](\xi) d\xi \\ &= \int_{\mathbb{R}^d} \int_0^1 [(x \cdot \xi + u)^+ - (-x \cdot \xi - u)^+] (2\mathrm{Re}[G](\xi) - \mathrm{Im}[G](\xi)) du d\xi. \end{aligned} \tag{9}$$

Combining (8) and (9) we obtain for all $x \in [-M, M]^d$

$$H(x) = \int_{\mathbb{R}^d} \int_{-\infty}^{\infty} (x \cdot \xi + u)^+ \alpha(\xi, u) du d\xi \tag{10}$$

with

$$\alpha(\xi, u) = -\mathbb{1}_{(-M\|\xi\|_1, 0]}(u) \mathrm{Re}[e^{-iu} G(\xi) + e^{iu} G(-\xi)] + \mathbb{1}_{[0,1]}(u) \tilde{g}(\xi) - \mathbb{1}_{[-1,0]}(u) \tilde{g}(-\xi)$$

for $\tilde{g}(\xi) = 2\mathrm{Re}[G](\xi) - \mathrm{Im}[G](\xi)$. Define for $(\xi, u) \in \mathbb{R}^d \times \mathbb{R}$ the function

$$f(\xi, u) = \frac{\alpha(\xi, u)}{\pi_{\mathrm{w}}(\xi) \pi_{\mathrm{b}}(u)}$$

and choose the random vector $W = (W_1, \ldots, W_N)$ as

$$W_i = \frac{1}{N} f(A_i, B_i), \quad i = 1, \ldots, N. \tag{11}$$

The estimate

$$|f(\xi, u)| \leq (\mathbb{1}_{(-M\|\xi\|_1, 0]}(u) + 4\mathbb{1}_{[-1,1]}(u)) \frac{|G(\xi)| + |G(-\xi)|}{\pi_{\mathrm{w}}(\xi) \pi_{\mathrm{b}}(u)} \tag{12}$$

then proves the claimed bound on $\|W_i\|_{L^\infty(\mathbb{P})}$ for $i = 1, \ldots, N$.

Step 2: We now use the representation (10) to prove (5) for the choice of W made in (11). To this end, first notice that for any $x \in [-M, M]^d$ we have by the choice of f and by the integral representation (10)

$$\mathbb{E}[f(A_i, B_i) \varrho(A_i \cdot x + B_i)] = \int_{\mathbb{R}^d} \int_{\mathbb{R}} \varrho(x \cdot \xi + u) \alpha(\xi, u) du d\xi = H(x).$$

Therefore, letting $U_{i,x} = f(A_i, B_i) \varrho(A_i \cdot x + B_i)$ for $i = 1, \ldots, N$ and $x \in [-M, M]^d$, we have

$$\mathbb{E}\left[\sup_{x \in [-M,M]^d} |H_W^{A,B}(x) - H(x)|\right] = \mathbb{E}\left[\sup_{x \in [-M,M]^d} \left|\frac{1}{N} \sum_{i=1}^N (U_{i,x} - \mathbb{E}[U_{i,x}])\right|\right]. \tag{13}$$

Let $\varepsilon_1, \dots, \varepsilon_N$ by i.i.d. Rademacher random variables which are independent of A and B. Symmetrization (see e.g. Boucheron et al., 2013) and (13) then yields

$$\mathbb{E}\left[\sup_{x\in[-M,M]^d} |H_W^{A,B}(x) - H(x)|\right] \leq 2\mathbb{E}\left[\sup_{x\in[-M,M]^d} \left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i U_{i,x}\right|\right]. \tag{14}$$

For $a = (a_1, \dots, a_N) \in \mathbb{R}^d \times \cdots \times \mathbb{R}^d$, $b \in \mathbb{R}^N$ we let $T_{a,b} = \{(|f(a_i, b_i)|[a_i \cdot x + b_i])_{i=1,\dots,N} \,|\, x \in [-M, M]^d\}$ and $\varrho_{a_i,b_i} = \operatorname{sign}(f(a_i, b_i))\varrho$ for $i = 1, \dots, N$. Then $T_{a,b} \subset \mathbb{R}^N$ is bounded, $|\varrho_{a_i,b_i}(s_1) - \varrho_{a_i,b_i}(s_2)| = |\varrho(s_1) - \varrho(s_2)| \leq |s_1 - s_2|$ for all $s_1, s_2 \in \mathbb{R}$, $i = 1, \dots, N$ and hence independence and Ledoux and Talagrand (2013, Theorem 4.12) yield

$$\begin{aligned}
&\mathbb{E}\left[\sup_{x\in[-M,M]^d} \left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i U_{i,x}\right|\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\sup_{t\in T_{a,b}} \left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i \varrho_{a_i,b_i}(t_i)\right|\right]\Bigg|_{(a,b)=(A,B)}\right] \\
&\leq 2\mathbb{E}\left[\mathbb{E}\left[\sup_{t\in T_{a,b}} \left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i t_i\right|\right]\Bigg|_{(a,b)=(A,B)}\right] \\
&= 2\mathbb{E}\left[\mathbb{E}\left[\sup_{x\in[-M,M]^d} \left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i f(a_i, b_i)[a_i \cdot x + b_i]\right|\right]\Bigg|_{(a,b)=(A,B)}\right].
\end{aligned} \tag{15}$$

Now for each a, b we use Jensen's inequality and the fact that $\mathbb{E}[\varepsilon_i \varepsilon_j] = \delta_{ij}$ to estimate

$$\begin{aligned}
&\mathbb{E}\left[\sup_{x\in[-M,M]^d} \left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i f(a_i, b_i)[a_i \cdot x + b_i]\right|\right] \\
&\leq \mathbb{E}\left[M\sqrt{d}\left\|\frac{1}{N}\sum_{i=1}^N \varepsilon_i f(a_i, b_i)a_i\right\| + \left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i f(a_i, b_i)b_i\right|\right] \\
&\leq M\sqrt{d}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N \varepsilon_i f(a_i, b_i)a_i\right\|^2\right]^{1/2} + \mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i f(a_i, b_i)b_i\right|^2\right]^{1/2} \\
&= \frac{M\sqrt{d}}{N}\left(\sum_{i=1}^N \|f(a_i, b_i)a_i\|^2\right)^{1/2} + \frac{1}{N}\left(\sum_{i=1}^N f(a_i, b_i)^2 b_i^2\right)^{1/2}.
\end{aligned} \tag{16}$$

Inserting this in (15) and using first Jensen's inequality and subsequently the fact that $(A_1, B_1), \dots, (A_N, B_N)$ are identically distributed yields

$$\begin{aligned}
&\mathbb{E}\left[\sup_{x\in[-M,M]^d}\left|\frac{1}{N}\sum_{i=1}^{N}\varepsilon_i U_{i,x}\right|\right] \\
&\leq \frac{2M\sqrt{d}}{N}\mathbb{E}\left[\left(\sum_{i=1}^{N}\|f(A_i,B_i)A_i\|^2\right)^{1/2}\right] + \frac{2}{N}\mathbb{E}\left[\left(\sum_{i=1}^{N} f(A_i,B_i)^2 B_i^2\right)^{1/2}\right] \\
&\leq \frac{2M\sqrt{d}}{\sqrt{N}}\mathbb{E}\left[\|f(A_1,B_1)A_1\|^2\right]^{1/2} + \frac{2}{\sqrt{N}}\mathbb{E}\left[f(A_1,B_1)^2 B_1^2\right]^{1/2}.
\end{aligned} \tag{17}$$

From the bound (12) we obtain

$$\begin{aligned}
&\mathbb{E}\left[f(A_1,B_1)^2\max(\|A_1\|^2, B_1^2)\right] \\
&\leq \int_{\mathbb{R}^d}\int_{\mathbb{R}}\left[(\mathbb{1}_{(-M\|\xi\|_1,0]}(u) + 4\mathbb{1}_{[-1,1]}(u))\frac{|G(\xi)|+|G(-\xi)|}{\pi_{\mathrm{w}}(\xi)\pi_{\mathrm{b}}(u)}\right]^2 \max(\|\xi\|^2, u^2)\pi_{\mathrm{w}}(\xi)\pi_{\mathrm{b}}(u)dud\xi \\
&\leq 2\int_{\mathbb{R}^d}\int_{\mathbb{R}}(\mathbb{1}_{(-M\|\xi\|_1,0]}(u) + 16\mathbb{1}_{[-1,1]}(u))\frac{(|G(\xi)|+|G(-\xi)|)^2}{\pi_{\mathrm{w}}(\xi)\pi_{\mathrm{b}}(u)}\max(\|\xi\|^2, u^2)dud\xi \\
&\leq \max(M^2,1)\int_{\mathbb{R}^d}\bar{F}(M\|\xi\|_1)\frac{(|G(\xi)|+|G(-\xi)|)^2}{\pi_{\mathrm{w}}(\xi)}\|\xi\|_1^2 d\xi \\
&\quad + 16\int_{\mathbb{R}^d}(\bar{F}(1)-\bar{F}(-1))\frac{(|G(\xi)|+|G(-\xi)|)^2}{\pi_{\mathrm{w}}(\xi)}\max(\|\xi\|^2,1)d\xi \\
&\leq I.
\end{aligned} \tag{18}$$

Combining this with (14) and (17) yields

$$\begin{aligned}
\mathbb{E}\left[\sup_{x\in[-M,M]^d}|H_W^{A,B}(x) - H(x)|\right] &\leq \frac{4(M\sqrt{d}+1)}{\sqrt{N}}\mathbb{E}\left[f(A_1,B_1)^2\max(\|A_1\|^2,B_1^2)\right]^{1/2} \\
&\leq \frac{4(M\sqrt{d}+1)\sqrt{I}}{\sqrt{N}},
\end{aligned} \tag{19}$$

which completes the proof. ■

Remark 3 *With some additional work the weight distributions in Theorem 1 could also be allowed to have compact support as in Gonon et al. (2023a, Theorem 1). However, in the results below (for instance in Theorem 7) such weight distributions would require much more restrictive assumptions on the unknown function H and thus we do not pursue this direction here.*

Remark 4 *The error bound* (5) *depends on the dimension d via the constant I in* (4)*. For a given H the dependence of I on d is determined by the choice of the densities π_w and π_b (based on which $\bar{F}$ is defined). Note that π_w is a density on $\mathbb{R}^d$ and so not only the*

decay behaviour of $\xi \mapsto \frac{\pi_w(\xi)}{\pi_w(0)}$*, but also the "normalizing constant"* $\pi_w(0)$ *determines how* I *depends on* d*. A key example of a situation in which* I *depends (at most) polynomially on* d *will be given in Theorem 7 below. Another example is as follows: let* π_w *be the density of a* $\mathcal{N}(0, \sigma^2 \mathbb{1}_d)$*-distribution, assume* $|G(\xi)| \leq C_1 \exp(-\frac{C_2}{4}\|\xi\|^2)$ *for some* $C_1 > 0, C_2 > \sigma^{-2}$*, and* π_b *has at most polynomial decay (see below). Then* $|\bar{F}(r)| \leq c(1+|r|^p)$ *for some* $p, c > 0$ *and one estimates*

$$\begin{aligned} I &\leq \tilde{c} d^{\frac{p}{2}+1} (2\pi\sigma^2)^{\frac{d}{2}} \int_{\mathbb{R}^d} \max(1, \|\xi\|^{p+2}) \exp\left(-\frac{C_2}{2}\|\xi\|^2 + \frac{\|\xi\|^2}{2\sigma^2}\right) d\xi \\ &\leq \tilde{c}_1 d^k (4\pi^2\sigma^2 (C_2 - \sigma^{-2})^{-1})^{\frac{d}{2}} \end{aligned} \tag{20}$$

for some constants $\tilde{c}$*,* $\tilde{c}_1$*,* k *not depending on* d*. If* $C_2 > \sigma^{-2} + 4\pi^2\sigma^2$*, then* $I \leq \tilde{c}_1 d^k$ *and so* I *is bounded polynomially in* d*.*

Corollary 5 *Assume that the hypotheses of Theorem 1 are satisfied. Then the random vector* W *from Theorem 1 also satisfies that for any probability measure* μ *on* $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ *which is supported in* $[-M, M]^d$ *we have that*

$$\mathbb{E}\left[\|H_W^{A,B} - H\|^2_{L^2(\mathbb{R}^d,\mu)}\right]^{1/2} \leq \frac{(\sqrt{d}M + 1)\sqrt{I}}{\sqrt{N}}. \tag{21}$$

Proof Using the same notation as in the proof of Theorem 1, we obtain from the proof of Theorem 1 and by Tonelli's theorem and independence that

$$\begin{aligned} \mathbb{E}\left[\|H_W^{A,B} - H\|^2_{L^2(\mathbb{R}^d,\mu)}\right] &= \mathbb{E}\left[\int_{\mathbb{R}^d} \left|\frac{1}{N}\sum_{i=1}^N U_{i,x} - \mathbb{E}[U_{i,x}]\right|^2 \mu(dx)\right] \\ &= \int_{\mathbb{R}^d} \frac{1}{N^2} \sum_{i=1}^N \mathbb{E}\left[|U_{i,x} - \mathbb{E}[U_{i,x}]|^2\right] \mu(dx) \\ &\leq \int_{\mathbb{R}^d} \frac{1}{N} \mathbb{E}\left[|f(A_1, B_1)\varrho(A_1 \cdot x + B_1)|^2\right] \mu(dx) \\ &\leq (\sqrt{d}M + 1)^2 \frac{\mathbb{E}\left[|f(A_1, B_1)\max(\|A_1\|, |B_1|)|^2\right]}{N}. \end{aligned}$$

Therefore, (18) yields the claimed bound. ∎

As a further corollary, the (quantitative) bounds above also imply a (qualitative) universal approximation result for random neural networks. The hidden weights in Corollary 6 are generated from an arbitrary distribution with polynomial tails. This complements Gonon et al. (2023a, Corollary 3), where the hidden weights are generated from a uniform distribution. The error in Corollary 6 is measured with respect to the supremum-norm on a compact set. An analogous result for $H \in L^p(\mathbb{R}^d, \mu)$ for a probability measure $\mu \in \mathbb{R}^d$, $p \in [1, \infty)$ and approximation error measured with respect to the norm on $L^p(\mathbb{R}^d, \mu)$ can be obtained similarly.

Corollary 6 *Let $H\colon \mathbb{R}^d \to \mathbb{R}$ be continuous. Assume that π_w and π_b are continuous, strictly positive and decay at most polynomially.*[1] *Then for any $M > 0$, $\varepsilon > 0$ there exist $N \in \mathbb{N}$ and an $\mathbb{R}^N$-valued, $\sigma(A, B)$-measurable random vector W such that*

$$\mathbb{E}\left[\sup_{x\in[-M,M]^d} |H_W^{A,B}(x) - H(x)|\right] < \varepsilon. \tag{22}$$

Proof Continuity of H implies that there exists $H_\varepsilon \in C_c^\infty(\mathbb{R}^d, \mathbb{R})$ such that

$$\sup_{x\in[-M,M]^d} |H(x) - H_\varepsilon(x)| < \frac{\varepsilon}{2}. \tag{23}$$

By assumption on π_{b} it follows that $\bar{F}(r) := 2\int_{-r}^0 \frac{1}{\pi_{\mathrm{b}}(s)} ds$ is finite for all $r \in \mathbb{R}$ and $\bar{F}$ grows at most polynomially. Let $\widehat{H_\varepsilon}(\xi) = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} e^{-ix\cdot\xi} H_\varepsilon(x) dx$ be the Fourier transform of H_ε. Then the fact that H_ε is a smooth function with compact support implies that $\widehat{H_\varepsilon}$ is a Schwartz function and Fourier inversion yields $H_\varepsilon(x) = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} e^{ix\cdot\xi} \widehat{H_\varepsilon}(\xi) d\xi$ for all $x \in \mathbb{R}^d$. The fact that $\widehat{H_\varepsilon}$ is a Schwartz function, at most polynomial growth of $\bar{F}$ and at most polynomial decay of π_{w} imply that

$$\int_{\mathbb{R}^d} [\bar{F}(M\|\xi\|_1)\|\xi\|_1^2 + \max(1, \|\xi\|^2)] \left(|\widehat{H_\varepsilon}(\xi)| + \frac{|\widehat{H_\varepsilon}(\xi)|^2 + |\widehat{H_\varepsilon}(-\xi)|^2}{\pi_{\mathrm{w}}(\xi)} \right) d\xi < \infty.$$

Theorem 1 hence proves that there exist $N \in \mathbb{N}$ and an $\mathbb{R}^N$-valued, $\sigma(A, B)$-measurable random vector W such that

$$\mathbb{E}\left[\sup_{x\in[-M,M]^d} |H_W^{A,B}(x) - H_\varepsilon(x)|\right] < \frac{\varepsilon}{2}. \tag{24}$$

Combining (23) and (24) completes the proof. ■

3. Random Neural Network Approximation Bounds

In this section we use random neural networks to approximate functions with a convolutional structure. In Section 3.1 we derive approximation error bounds with explicit dependence on the dimension d. These results are then applied in Section 3.2 in the context of certain exponential Lévy models, which include the Black-Scholes model as a special case.

3.1 Bounds for Convolutional Functions

Consider a function $H\colon \mathbb{R}^d \to \mathbb{R}$ given by $H(x) = \mathbb{E}[\Phi(x + V)]$ for an $\mathbb{R}^d$-valued random vector V and a function $\Phi\colon \mathbb{R}^d \to \mathbb{R}$. Assume that the characteristic function of V satisfies the following bound: there exists $C > 0$ such that

$$|\mathbb{E}[e^{i\xi\cdot V}]| \le \exp(-C\|\xi\|^2) \quad \text{for all } \xi \in \mathbb{R}^d. \tag{25}$$

1. That is, there exist polynomials $p_{\mathrm{w}}, p_{\mathrm{b}}\colon \mathbb{R} \to (0, \infty)$ such that $1 \le p_{\mathrm{w}}(\|\xi\|)\pi_{\mathrm{w}}(\xi)$ and $1 \le p_{\mathrm{b}}(z)\pi_{\mathrm{b}}(z)$ for all $\xi \in \mathbb{R}^d$, $z \in \mathbb{R}$.

Examples of functions H of this type include expectations (respectively option prices) and associated solutions to PDEs in (exponential) Lévy models with non-degenerate Gaussian component, see Section 3.2 below.

We now approximate H by a random neural network $H_W^{A,B}$ and analyze the approximation error. As above the randomly generated hidden weights A, B are not trainable and the goal is to find W such that the expected uniform approximation error $\mathbb{E}[\|H_W^{A,B} - H\|_{L^\infty([-M,M]^d)}]$ is small. The output weight vector W may be chosen depending on A, B, i.e., it is a $\sigma(A,B)$-measurable random variable.

Our goal is to obtain approximation error bounds in which the dependence on the dimension d is fully explicit. To achieve this we need more specific assumptions on the distributions from which the hidden weights A and B are drawn. Recall that π_{b} denotes the Lebesgue-density of B_1 and π_{w} denotes the density of A_1. We will assume below that π_{w} is the density of a multivariate t-distribution $t_\nu(0, \mathbb{1}_d)$ for some $\nu > 1$ and that π_{b} has *at most polynomial decay*, that is, there exists a polynomial $p_{\mathrm{b}} \colon \mathbb{R} \to (0,\infty)$ such that

$$1 \leq p_{\mathrm{b}}(z)\pi_{\mathrm{b}}(z) \quad \text{for all } z \in \mathbb{R}. \tag{26}$$

This hypothesis is satisfied, for instance, by Student's t-distribution.

These assumptions allow us to obtain explicit control of the normalizing constant of the weight distribution π_{w}.

Theorem 7 *Let $C > \frac{1}{2^{3/2}\pi}$ and let $\nu > 1$. Suppose $A_1 \sim t_\nu(0, \mathbb{1}_d)$ and B_1 has density π_b satisfying (26). Then there exist $k \in \mathbb{N}$ and an absolute constant $C_{app} > 0$ such that for any $H \colon \mathbb{R}^d \to \mathbb{R}$ of the form $H(x) = \mathbb{E}[\Phi(x+V)]$ with $\Phi \in L^1(\mathbb{R}^d)$ and V satisfying (25) the following random neural network approximation result holds: there exists an $\mathbb{R}^N$-valued, $\sigma(A,B)$-measurable random vector W such that*

$$\mathbb{E}\left[\sup_{x\in[-M,M]^d} |H_W^{A,B}(x) - H(x)|\right] \leq \frac{C_{app}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu+d)^{k+3}}{\sqrt{N}}. \tag{27}$$

The constant k only depends on π_b and the constant C_{app} depends on ν, π_b, C, M, but it does not depend on d, N or H.

Moreover,

$$\|W_i\|_{L^\infty(\mathbb{P})} \leq \frac{C_{wgt}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu+d)^{2k+\frac{1}{2}}}{N} \tag{28}$$

for $i = 1, \dots, N$, where the constant $C_{wgt} > 0$ depends on ν, π_b, C, M, but it does not depend on d, N or H.

Remark 8 *In addition to the uniform bound in (27) the proof of Theorem 7 also shows that for any probability measure μ on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ supported in $[-M,M]^d$ we have*

$$\mathbb{E}\left[\|H_W^{A,B} - H\|^2_{L^2(\mathbb{R}^d,\mu)}\right]^{1/2} \leq \frac{C_{app}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu+d)^{k+3}}{\sqrt{N}}. \tag{29}$$

This follows directly by using Corollary 5 instead of Theorem 1 in (37) below.

Remark 9 *Hypothesis* (25) *in Theorem 7 is employed in the proof in order to guarantee that the constant in the error bound does not grow exponentially in the dimension d. In low-dimensional situations this behaviour may not be required and hence* (25) *could be replaced by the weaker hypothesis $|\mathbb{E}[e^{i\xi\cdot V}]| \leq \exp(-C\|\xi\|^\alpha)$ for some $C > 0$, $\alpha > 0$ or even by the assumption that $|\mathbb{E}[e^{i\xi\cdot V}]| \leq C(1+\|\xi\|)^{-\beta}$ for some $C > 0$ and sufficiently large $\beta > 0$ (depending on ν and π_b). In this situation the error bound* (27) *is still valid with a different constant C_{app} and an additional factor which may not necessarily be polynomial in d.*

Proof Let $H\colon \mathbb{R}^d \to \mathbb{R}$ be of the form $H(x) = \mathbb{E}[\Phi(x+V)]$ with $\Phi \in L^1(\mathbb{R}^d)$ and V satisfying (25). We verify that H satisfies the hypotheses of Theorem 1 and derive a bound for the constant I in (4) with the claimed properties.

For $f \in L^1(\mathbb{R}^d)$ we denote by $\hat{f}$ the Fourier transform of f given for all $\xi \in \mathbb{R}^d$ by $\hat{f}(\xi) = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} e^{-ix\cdot\xi} f(x)dx$. By (25) and Sato (1999, Proposition 2.5(xii)) the random variable $-V$ has a bounded Lebesgue-density p_{-V}. Thus, we can write $H(x) = \int_{\mathbb{R}^d} \Phi(x-y)p_{-V}(y)dy = (\Phi * p_{-V})(x)$. The convolution theorem (see for instance Amann and Escher (2009, Theorem X.9.16)) hence shows that $\hat{H}(\xi) = (2\pi)^{\frac{d}{2}}\hat{\Phi}(\xi)\widehat{p_{-V}}(\xi)$. Combining this with $\widehat{p_{-V}}(\xi) = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} e^{-ix\cdot\xi} p_{-V}(x)dx = (2\pi)^{-\frac{d}{2}}\mathbb{E}[e^{-i\xi\cdot(-V)}]$ and (25) we obtain that $\hat{H}$ is integrable. The Fourier inversion theorem (see for instance Amann and Escher (2009, Theorem X.9.12)) therefore yields for all $x \in \mathbb{R}^d$ that

$$H(x) = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} e^{i\xi\cdot x}\hat{H}(\xi)d\xi = (2\pi)^{-\frac{d}{2}} \int_{\mathbb{R}^d} e^{i\xi\cdot x}\hat{\Phi}(\xi)\mathbb{E}[e^{i\xi\cdot V}]d\xi. \tag{30}$$

Hence, the representation (2) holds for all $x \in \mathbb{R}^d$ with $G(\xi) = (2\pi)^{-\frac{d}{2}}\hat{\Phi}(\xi)\mathbb{E}[e^{i\xi\cdot V}]$, $\xi \in \mathbb{R}^d$. Condition (3) is satisfied, since (25) implies

$$\int_{\mathbb{R}^d} \max(1,\|\xi\|^2)|G(\xi)|d\xi \leq (2\pi)^{-d}\|\Phi\|_{L^1(\mathbb{R}^d)} \int_{\mathbb{R}^d} \max(1,\|\xi\|^2)\exp(-C\|\xi\|^2)d\xi < \infty.$$

Denote by $k \in \mathbb{N}$ the degree of p_{b}, then there exist $a_0, \ldots, a_k \in \mathbb{R}$ such that $p_{\mathrm{b}}(s) = \sum_{l=0}^k a_l s^l$ for all $s \in \mathbb{R}$. Then $|p_{\mathrm{b}}(s)| \leq (k+1)\max_l\{|a_l|\}\max(1,|s|^k) \leq C_{\mathrm{b}}(1+s^{2k})$ for all $s \in \mathbb{R}$, where $C_{\mathrm{b}} = (k+1)\max_l\{|a_l|\}$. Consequently, we may use (26) to estimate for any $r \geq 0$

$$\bar{F}(r) = 2\int_{-r}^0 \frac{1}{\pi_{\mathrm{b}}(s)}ds \leq 2\int_{-r}^0 p_{\mathrm{b}}(s)ds \leq 2C_{\mathrm{b}}(r + \frac{r^{2k+1}}{2k+1}) < \infty$$

and for $r < 0$ analogously $|\bar{F}(r)| = 2\int_0^{-r} \frac{1}{\pi_{\mathrm{b}}(s)}ds \leq -2C_{\mathrm{b}}(r + \frac{r^{2k+1}}{2k+1}) < \infty$. Therefore, $\bar{F}(1) - \bar{F}(-1) \leq 8C_{\mathrm{b}}$ and we can now use the comparison $\|\cdot\|_1 \leq \sqrt{d}\|\cdot\|$ on $\mathbb{R}^d$ to estimate the constant I in (4) as

$$\begin{aligned}
I &\leq 2C_{\mathrm{b}}c_{M,1}\int_{\mathbb{R}^d}\left[M\|\xi\|_1^3 + \frac{\|\xi\|_1^2(M\|\xi\|_1)^{2k+1}}{2k+1} + 4\max(1,\|\xi\|^2)\right]\frac{(|G(\xi)|+|G(-\xi)|)^2}{\pi_{\mathrm{w}}(\xi)}d\xi \\
&\leq 2C_{\mathrm{b}}c_{M,2}d^{k+\frac{3}{2}}\int_{\mathbb{R}^d}\left[\|\xi\|^3 + \|\xi\|^{2k+3} + \max(1,\|\xi\|^2)\right]\frac{(|G(\xi)|+|G(-\xi)|)^2}{\pi_{\mathrm{w}}(\xi)}d\xi \\
&\leq 6C_{\mathrm{b}}c_{M,2}d^{k+\frac{3}{2}}\int_{\mathbb{R}^d}\max(1,\|\xi\|^{2k+3})\frac{(\hat{\Phi}(\xi)\mathbb{E}[e^{i\xi\cdot V}] + \hat{\Phi}(-\xi)\mathbb{E}[e^{-i\xi\cdot V}])^2}{(2\pi)^d\pi_{\mathrm{w}}(\xi)}d\xi \\
&\leq C_I d^{k+\frac{3}{2}}\|\Phi\|_{L^1(\mathbb{R}^d)}^2\int_{\mathbb{R}^d}\max(1,\|\xi\|^{2k+3})\frac{\exp(-2C\|\xi\|^2)}{(2\pi)^{2d}\pi_{\mathrm{w}}(\xi)}d\xi
\end{aligned}$$

with $c_{M,1} = \max(M^2, 16)$, $c_{M,2} = c_{M,1}\max(M^{2k+1}, 4)$, $C_I = 24C_{\mathrm{b}}c_{M,2}$.

We now insert the density $\pi_{\mathrm{w}}(x) = \frac{\Gamma((\nu+d)/2)}{\Gamma(\nu/2)\nu^{d/2}\pi^{d/2}}(1+\nu^{-1}\|x\|^2)^{-(\nu+d)/2}$ and use the estimate $(\nu+\|x\|^2)^p \leq 2^{p-1}(\nu^p + \|x\|^{2p})$ for $p \geq 1$ to obtain

$$\begin{aligned} I &\leq C_I d^{k+\frac{3}{2}}\|\Phi\|^2_{L^1(\mathbb{R}^d)} \int_{\mathbb{R}^d} \max(1, \|\xi\|^{2k+3})(\nu+\|\xi\|^2)^{(\nu+d)/2} \frac{\exp(-2C\|\xi\|^2)\Gamma(\frac{\nu}{2})\nu^{-\frac{\nu}{2}}\pi^{d/2}}{(2\pi)^{2d}\Gamma(\frac{\nu+d}{2})} d\xi \\ &\leq C_I d^{k+\frac{3}{2}}\|\Phi\|^2_{L^1(\mathbb{R}^d)}\Gamma\left(\frac{\nu}{2}\right)\nu^{-\nu/2} \int_{\mathbb{R}^d} (\nu+\|\xi\|^2)^{(2k+3+\nu+d)/2} \frac{\exp(-2C\|\xi\|^2)\pi^{d/2}}{(2\pi)^{2d}\Gamma(\frac{\nu+d}{2})} d\xi \\ &\leq \frac{\tilde{C}_I d^{k+\frac{3}{2}}\|\Phi\|^2_{L^1(\mathbb{R}^d)}}{(2\pi)^{3d/2}\Gamma(\frac{\nu+d}{2})} \int_{\mathbb{R}^d} (\nu^{(\tilde{\nu}+d)/2} + \|\xi\|^{\tilde{\nu}+d}) \exp(-2C\|\xi\|^2) d\xi \end{aligned} \tag{31}$$

with $\tilde{\nu} = 2k+3+\nu$, $\tilde{C}_I = 2^{(\tilde{\nu}/2)-1}C_I\Gamma(\frac{\nu}{2})\nu^{-\nu/2}$. Denote by X_C a random variable with a $\mathcal{N}(0, \frac{1}{4C}\mathbb{1}_d)$-distribution. Then the last line in (31) can be rewritten in terms of X_C, yielding

$$\begin{aligned} I &\leq \frac{\tilde{C}_I d^{k+\frac{3}{2}}\|\Phi\|^2_{L^1(\mathbb{R}^d)}}{(2\pi)^d\Gamma(\frac{\nu+d}{2})(2^dC^{d/2})} \int_{\mathbb{R}^d} (\nu^{(\tilde{\nu}+d)/2} + \|\xi\|^{\tilde{\nu}+d}) \frac{\exp(-2C\|\xi\|^2)}{(2\pi)^{d/2}(4C)^{-d/2}} d\xi \\ &= \frac{\tilde{C}_I d^{k+\frac{3}{2}}\|\Phi\|^2_{L^1(\mathbb{R}^d)}}{(2\pi)^d\Gamma(\frac{\nu+d}{2})(2^dC^{d/2})} \left[\nu^{(\tilde{\nu}+d)/2} + \mathbb{E}[\|X_C\|^{\tilde{\nu}+d}]\right]. \end{aligned} \tag{32}$$

On the other hand, $\mathbb{E}[\|X_C\|^{\tilde{\nu}+d}] = \mathbb{E}[\|(2\sqrt{C})^{-1}Z\|^{\tilde{\nu}+d}] = (2\sqrt{C})^{-(\tilde{\nu}+d)}\mathbb{E}[(\|Z\|^2)^{\frac{\tilde{\nu}+d}{2}}]$ where Z is a d-dimensional standard normal random vector. Hence, $\|Z\|^2$ has a $\chi^2(d)$-distribution and thus

$$\begin{aligned} \mathbb{E}[(\|Z\|^2)^{(\tilde{\nu}+d)/2}] &= 2^{-d/2}[\Gamma(d/2)]^{-1}\int_0^\infty x^{\frac{\tilde{\nu}+2d}{2}-1}e^{-x/2}dx = \frac{2^{(\tilde{\nu}+2d)/2}\Gamma((\tilde{\nu}+2d)/2)}{2^{d/2}\Gamma(d/2)} \\ &= \frac{2^{(\tilde{\nu}+d)/2}\Gamma((\tilde{\nu}+2d)/2)}{\Gamma(d/2)}. \end{aligned}$$

Combining this with (32) and the upper and lower bounds for the gamma function (see, e.g., Gonon et al. 2021, Lemma 2.4) we obtain

$$I \leq \frac{\tilde{C}_I d^{k+\frac{3}{2}}\|\Phi\|^2_{L^1(\mathbb{R}^d)}}{(2\pi)^d\Gamma(\frac{\nu+d}{2})(2^dC^{d/2})} \left[\nu^{(\tilde{\nu}+d)/2} + (2\sqrt{C})^{-(\tilde{\nu}+d)}\frac{2^{(\tilde{\nu}+d)/2}\Gamma((\tilde{\nu}+2d)/2)}{\Gamma(d/2)}\right] \tag{33}$$

$$\begin{aligned} &\leq \frac{\tilde{C}_I d^{k+\frac{3}{2}}\|\Phi\|^2_{L^1(\mathbb{R}^d)}}{(2\pi)^d(2^dC^{d/2})}(\frac{\nu+d}{4\pi})^{\frac{1}{2}}(\frac{2e}{\nu+d})^{\frac{\nu+d}{2}}\left[\nu^{\frac{\tilde{\nu}+d}{2}} + \frac{e}{(2C)^{\frac{\tilde{\nu}+d}{2}}}(\frac{2e}{d})^{\frac{d}{2}}(\frac{d}{\tilde{\nu}+2d})^{\frac{1}{2}}(\frac{\tilde{\nu}+2d}{2e})^{\frac{\tilde{\nu}+2d}{2}}\right] \\ &\leq \frac{\tilde{C}_I d^{k+\frac{3}{2}}\|\Phi\|^2_{L^1(\mathbb{R}^d)}\sqrt{\nu+d}}{(16\pi^2C)^{\frac{\tilde{\nu}+d}{2}}}(\frac{2e}{\nu+d})^{\frac{\tilde{\nu}+d-2k-3}{2}}(4\pi\sqrt{C})^{\tilde{\nu}}\left[\nu^{\frac{\tilde{\nu}+d}{2}} + \frac{(2e)^{\frac{d}{2}}}{(2C)^{\frac{\tilde{\nu}+d}{2}}d^{\frac{d}{2}}}(\frac{\tilde{\nu}+2d}{2e})^{\frac{\tilde{\nu}+2d}{2}}\right] \\ &\leq \bar{C}_I d^{k+\frac{3}{2}}\|\Phi\|^2_{L^1(\mathbb{R}^d)}(\nu+d)^{k+2}\left[(\frac{e\nu}{8\pi^2C(\nu+d)})^{\frac{\tilde{\nu}+d}{2}} + (\frac{\tilde{\nu}+2d}{32\pi^2C^2(\nu+d)})^{\frac{\tilde{\nu}+d}{2}}(\frac{\tilde{\nu}+2d}{d})^{\frac{d}{2}}\right] \\ &\leq \bar{C}_I\|\Phi\|^2_{L^1(\mathbb{R}^d)}(\nu+d)^{2k+\frac{7}{2}}\left[(\frac{e\nu}{8\pi^2C(\nu+d)})^{\frac{\tilde{\nu}+d}{2}} + (\frac{\tilde{\nu}+2d}{32\pi^2C^2(\nu+d)})^{\frac{\tilde{\nu}}{2}}(\frac{(\tilde{\nu}+2d)^2}{32\pi^2C^2(\nu+d)d})^{\frac{d}{2}}\right] \end{aligned}$$

with $\bar{C}_I = \tilde{C}_I(2e)^{-\frac{2k+3}{2}}(16\pi^2 C)^{\frac{\tilde{\nu}}{2}}$. Now clearly

$$C_1 := \sup_{m\in\mathbb{N}} \left\{ \left(\frac{e\nu}{8\pi^2 C(\nu+m)} \right)^{\frac{\tilde{\nu}+m}{2}} \right\} < \infty \tag{34}$$

and

$$\begin{aligned} C_2 :&= \sup_{m\in\mathbb{N}} \left\{ \left(\frac{(\tilde{\nu}+2m)^2}{32\pi^2 C^2(\nu+m)m} \right)^{\frac{m}{2}} \right\} = \sup_{m\in\mathbb{N}} \left\{ \left(\frac{(\frac{\tilde{\nu}}{m}+2)^2}{32\pi^2 C^2(\frac{\nu}{m}+1)} \right)^{\frac{m}{2}} \right\} \\ &\leq \sup_{m\in\mathbb{N}} \left\{ \left(\frac{(\frac{\tilde{\nu}}{m}+2)^2}{32\pi^2 C^2} \right)^{\frac{m}{2}} \right\} < \infty, \end{aligned} \tag{35}$$

because $C^2 > \frac{1}{8\pi^2}$ and hence $m_0 := \frac{\tilde{\nu}}{2(\sqrt{8}\pi C - 1)} > 0$ and $(\frac{\tilde{\nu}}{m}+2)^2 < 32\pi^2 C^2$ for all $m \in \mathbb{N}$ with $m > m_0$. Combining this with (33) we have therefore proved that

$$\begin{aligned} I &\leq \bar{C}_I \|\Phi\|^2_{L^1(\mathbb{R}^d)} (\nu+d)^{2k+\frac{7}{2}} \left[C_1 + \left(\frac{\tilde{\nu}+2d}{32\pi^2 C^2(\nu+d)} \right)^{\frac{\tilde{\nu}}{2}} C_2 \right] \\ &\leq \bar{C}_I \|\Phi\|^2_{L^1(\mathbb{R}^d)} (\nu+d)^{2k+\frac{7}{2}} \left[C_1 + \left(\frac{\tilde{\nu}+2}{32\pi^2 C^2} \right)^{\frac{\tilde{\nu}}{2}} C_2 \right] \\ &= C_3 \|\Phi\|^2_{L^1(\mathbb{R}^d)} (\nu+d)^{2k+\frac{7}{2}} \end{aligned} \tag{36}$$

with $C_3 = \bar{C}_I(C_1 + (\frac{\tilde{\nu}+2}{32\pi^2 C^2})^{\frac{\tilde{\nu}}{2}} C_2)$.

Altogether, the hypotheses of Theorem 1 are satisfied and hence there exists an $\mathbb{R}^N$-valued, $\sigma(A,B)$-measurable random vector W such that the error bound (5) holds. Inserting (36) yields

$$\begin{aligned} \mathbb{E}\left[\sup_{x\in[-M,M]^d} |H_W^{A,B}(x) - H(x)| \right] &\leq \frac{4(M+1)\sqrt{d}\sqrt{I}}{\sqrt{N}} \\ &\leq \frac{4(M+1)\sqrt{C_3}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu+d)^{k+3}}{\sqrt{N}}. \end{aligned} \tag{37}$$

Hence, the L^∞-error estimate (27) follows with

$$C_{\text{app}} = 4(M+1)\sqrt{C_3} = 4(M+1)(\bar{C}_I)^{1/2} \left[C_1 + \left(\frac{\tilde{\nu}+2}{32\pi^2 C^2} \right)^{\frac{\tilde{\nu}}{2}} C_2 \right]^{1/2}$$

where we recall that $\bar{C}_I = 24 C_{\text{b}} \max(M^2, 16) \max(M^{2k+1}, 4) 2^{(\tilde{\nu}/2)-1} \Gamma(\frac{\nu}{2}) \nu^{-\nu/2} (2e)^{-\frac{2k+3}{2}} (16\pi^2 C)^{\frac{\tilde{\nu}}{2}}$, $\tilde{\nu} = 2k+3+\nu$, the constants C_{b} and k only depend on p_{b} and C_1, C_2 are given by (34) and (35), respectively.

To prove the upper bound on W we insert the bound from Theorem 1, then (25) and (26) can be used to estimate similarly as before for $i = 1, \dots, N$

$$\begin{aligned}
\|W_i\|_{L^\infty(\mathbb{P})} &\leq \frac{1}{N} \sup_{(u,\xi)\in\mathbb{R}\times\mathbb{R}^d} (\mathbb{1}_{[-M\|\xi\|_1,0]}(u) + 4\mathbb{1}_{[-1,1]}(u)) \frac{|G(\xi)| + |G(-\xi)|}{\pi_{\mathrm{b}}(u)\pi_{\mathrm{w}}(\xi)} \\
&\leq \frac{5}{N} \sup_{\xi\in\mathbb{R}^d} \frac{2(2\pi)^{-d}\|\Phi\|_{L^1(\mathbb{R}^d)} \exp(-C\|\xi\|^2) C_{\mathrm{b}}(1 + \max(1, M\|\xi\|_1)^{2k})}{\pi_{\mathrm{w}}(\xi)} \\
&\leq \frac{\tilde{C}_{\mathrm{wgt}} d^k \|\Phi\|_{L^1(\mathbb{R}^d)}}{N} \sup_{\xi\in\mathbb{R}^d} \frac{\exp(-C\|\xi\|^2)(2 + M^{2k}\|\xi\|^{2k})(\nu + \|\xi\|^2)^{(\nu+d)/2}}{2^{\frac{d}{2}}\Gamma((\nu+d)/2)(2\pi)^{\frac{d}{2}}} \\
&\leq \frac{\tilde{C}_{\mathrm{wgt}} d^k \|\Phi\|_{L^1(\mathbb{R}^d)} \max(2, M^{2k})}{N} \max_{r\geq 0} \frac{\exp(-Cr)(\nu + r)^{(\nu+d+2k)/2}}{2^{\frac{d}{2}}\Gamma((\nu+d)/2)(2\pi)^{\frac{d}{2}}} \\
&= \frac{\tilde{C}_{\mathrm{wgt}} d^k \|\Phi\|_{L^1(\mathbb{R}^d)} \max(2, M^{2k})}{N} e^{\nu C} \frac{((\nu + d + 2k)/(2Ce))^{(\nu+d+2k)/2}}{2^{\frac{d}{2}}\Gamma((\nu+d)/2)(2\pi)^{\frac{d}{2}}} \\
&\leq \frac{\bar{C}_{\mathrm{wgt}} \|\Phi\|_{L^1(\mathbb{R}^d)} (\nu + d)^{2k+\frac{1}{2}}}{N} \left(\frac{\nu + d + 2k}{4\pi C(\nu + d)}\right)^{(\nu+d+2k)/2}
\end{aligned}$$

with $\tilde{C}_{\mathrm{wgt}} = 10C_{\mathrm{b}}\Gamma(\nu/2)\nu^{-\nu/2}$, $\bar{C}_{\mathrm{wgt}} = \tilde{C}_{\mathrm{wgt}} \max(2, M^{2k}) e^{\nu C}(2e)^{-k}(4\pi)^{(\nu+2k-1)/2}$. In the last two steps we used that the maximum is attained for $\nu + r = (\nu + d + 2k)/(2C)$ and we applied the lower bound for the gamma function as in (33). The hypothesis $8\pi^2C^2 > 1$ implies $4\pi C > 1$ and therefore

$$C_4 := \sup_{m\in\mathbb{N}} \left\{ \left(\frac{\nu + m + 2k}{4\pi C(\nu + m)}\right)^{(\nu+m+2k)/2} \right\} < \infty$$

by a similar reasoning as used to argue that C_2 in (35) is finite. Hence, the bound on $\|W_i\|_{L^\infty(\mathbb{P})}$ follows with $C_{\mathrm{wgt}} = \bar{C}_{\mathrm{wgt}} C_4$. This completes the proof. ■

Remark 10 *In general (when $C > \frac{1}{2^{3/2}\pi}$ is not necessarily satisfied) the proof (with constants C_2 and C_4 defined slightly differently) still yields the bounds* (27), (28) *with additional factors $(8\pi^2C^2)^{-\frac{d}{4}}$ and $(4\pi C)^{-\frac{d}{2}}$, respectively. Thus, in the case $C \leq \frac{1}{2^{3/2}\pi}$ the bounds have constants polynomial in d provided that $\|\Phi\|_{L^1(\mathbb{R}^d)}(2^{\frac{3}{2}}\pi C)^{-\frac{d}{2}}$ is at most polynomial in d.*

We now show that an analogous approximation result holds when the assumption $\Phi \in L^1(\mathbb{R}^d)$ is replaced by the assumptions that Φ satisfies a Lipschitz-condition and V admits certain moments.

Here we call $\psi\colon \mathbb{R}^d \to \mathbb{R}^d$ increasing if for any $x, y \in \mathbb{R}^d$ with $x_i \leq y_i$ for all $i = 1, \dots, d$ it holds that $\psi_i(x) \leq \psi_i(y)$ for all $i = 1, \dots, d$. Furthermore, we denote $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^d$.

Proposition 11 *Let $C, C_{Lip} > 0$ with $C^2 > \frac{1}{8\pi^2}$ and let $\nu > 1$. Suppose $A_1 \sim t_\nu(0, \mathbb{1}_d)$ and B_1 has density π_b satisfying* (26). *Let $\psi\colon \mathbb{R}^d \to \mathbb{R}^d$ be increasing and measurable. Let $H\colon \mathbb{R}^d \to \mathbb{R}$ be of the form $H(x) = \mathbb{E}[\Phi(x + V)]$ for Φ satisfying*

$$|\Phi(x) - \Phi(y)| \leq C_{Lip}\|\psi(x) - \psi(y)\|, \quad x, y \in \mathbb{R}^d \tag{38}$$

and V satisfying (25), $\mathbb{E}[\|\psi(M\mathbf{1}+V)\|^2]<\infty$. Then for any $R>0$ there exists an $\mathbb{R}^N$-valued, $\sigma(A,B)$-measurable random vector W such that

$$\mathbb{E}\left[\sup_{x\in B_M(0)}|H_W^{A,B}(x)-H(x)|\right]\leq\frac{C_{app}\mathcal{I}(R)(\nu+d)^{k+3}}{\sqrt{N}}+C_{mom}\mathbb{P}(\|V\|>R)^{1/2},\tag{39}$$

where $\mathcal{I}(R)=\int_{\mathbb{R}^d}|\Phi(x)|\mathbb{1}_{\{\|x\|\leq M+R\}}dx$, $C_{mom}=C_{Lip}(\mathbb{E}[\|\psi(M\mathbf{1}+V)\|^2]^{1/2}+\|\psi(0)\|)+|\Phi(0)|$ and $k\in\mathbb{N}$, $C_{app}>0$ are as in Theorem 7.

Proof Let $R>0$ and denote $\Phi^R(x)=\Phi(x)\mathbb{1}_{\{\|x\|\leq M+R\}}$. Set $\bar{H}^R(x)=\mathbb{E}[\Phi^R(x+V)]$. Then for $x\in B_M(0)$ we estimate

$$\begin{aligned}
&|\bar{H}^R(x)-H(x)|\\
&\leq\mathbb{E}[|\Phi(x+V)\mathbb{1}_{\{\|x+V\|\leq M+R\}}-\Phi(x+V)|]\\
&=\mathbb{E}[\mathbb{1}_{\{\|x+V\|>M+R\}}|\Phi(x+V)|]\\
&\leq C_{\mathrm{Lip}}\mathbb{E}[\mathbb{1}_{\{\|x+V\|>M+R\}}\|\psi(x+V)-\psi(0)\|]+|\Phi(0)|\mathbb{P}(\|x+V\|>M+R)\\
&\leq C_{\mathrm{Lip}}\mathbb{E}[\mathbb{1}_{\{\|x+V\|>M+R\}}\|\psi(M\mathbf{1}+V)\|]+(C_{\mathrm{Lip}}\|\psi(0)\|+|\Phi(0)|)\mathbb{P}(\|x+V\|>M+R)\\
&\leq C_{\mathrm{Lip}}\mathbb{E}[\mathbb{1}_{\{\|V\|>R\}}\|\psi(M\mathbf{1}+V)\|]+(C_{\mathrm{Lip}}\|\psi(0)\|+|\Phi(0)|)\mathbb{P}(\|V\|>R)\\
&\leq C_{\mathrm{Lip}}\mathbb{P}(\|V\|>R)^{1/2}\mathbb{E}[\|\psi(M\mathbf{1}+V)\|^2]^{1/2}+(C_{\mathrm{Lip}}\|\psi(0)\|+|\Phi(0)|)\mathbb{P}(\|V\|>R)^{1/2}.
\end{aligned}$$

The truncated function Φ^R is integrable and

$$\|\Phi^R\|_{L^1(\mathbb{R}^d)}=\int_{\mathbb{R}^d}|\Phi(x)|\mathbb{1}_{\{\|x\|\leq M+R\}}dx=\mathcal{I}(R).$$

Therefore, the result follows from Theorem 7 and the triangle inequality. ■

Remark 12 *Let us now explain how Proposition 11 could be applied. In the case of exponential Lévy models we would choose $\Phi(x)=\varphi(\exp(x))$ for $\varphi\colon\mathbb{R}^d\to\mathbb{R}$. Hence, if φ is C_{Lip}-Lipschitz-continuous, then (38) is satisfied with $\psi(x)=(\exp(x_1),\dots,\exp(x_d))$ for $x\in\mathbb{R}^d$. Consequently, if we choose $R=\frac{1}{\alpha}\log(N)$ for some $\alpha>1$, then*

$$\begin{aligned}
\mathcal{I}(R)&=\int_{\mathbb{R}^d}|\Phi(x)|\mathbb{1}_{\{\|x\|\leq M+R\}}dx\leq\int_{\mathbb{R}^d}c(1+\|\exp(x)\|)\mathbb{1}_{\{\|x\|\leq M+R\}}dx\\
&\leq\int_{\mathbb{R}^d}c(1+d^{\frac{1}{2}}\exp(M+R))\mathbb{1}_{\{\|x\|\leq M+R\}}dx\\
&=c(1+d^{\frac{1}{2}}\exp(M+R))\mathrm{Vol}(B_{M+R}(0))\\
&\leq c(1+d^{\frac{1}{2}}e^M N^{\frac{1}{\alpha}})(d\pi)^{-1/2}\left(\frac{2\pi e}{d}\right)^{d/2}\left(M+\frac{\log(N)}{\alpha}\right)^d\\
&\leq\tilde{c}N^{\frac{1}{\alpha}}
\end{aligned}$$

with $c=\max(C_{Lip},|\varphi(0)|)$, $\tilde{c}=2c\max(1,e^M)\pi^{-1/2}$ and where the last step holds if the number of nodes satisfies the condition $N\leq\exp(\alpha[d^{1/2}(2\pi e)^{-1/2}-M])$ (which is, however,

exponential in d). Furthermore,

$$\begin{aligned} C_{mom} &= C_{Lip}\mathbb{E}[\|\exp(M\mathbf{1}+V)\|^2]^{1/2} + C_{Lip}\|\mathbf{1}\| + |\varphi(\mathbf{1})| \\ &\leq C_{Lip}e^M \left(\sum_{i=1}^d \mathbb{E}[\exp(2V_i)]\right)^{1/2} + d^{1/2}C_{Lip} + c(1+d^{1/2}) \end{aligned}$$

is finite under exponential moment hypotheses on V. Therefore, from (39) *and Markov's inequality we obtain*

$$\mathbb{E}\left[\sup_{x\in B_M(0)} |H_W^{A,B}(x) - H(x)|\right] \leq \frac{\tilde{c}C_{app}(\nu+d)^{k+3} + C_{mom}\mathbb{E}[\exp(\alpha\|V\|)]^{1/2}}{N^{\frac{1}{2}-\frac{1}{\alpha}}}. \tag{40}$$

3.2 Bounds for Non-degenerate Lévy Models

In this section we apply Theorem 7 to prove that random neural networks are capable of overcoming the curse of dimensionality in the numerical approximation of solutions to partial (integro-)differential equations (also referred to as (non-local) PDEs) associated to exponential Lévy models with a non-degenerate Gaussian component. This includes the Black-Scholes PDE as a special case. We refer to Cont and Tankov (2004), Eberlein and Kallsen (2019) for background on exponential Lévy models and their applications in financial modelling and, e.g., to Sato (1999) for an extensive treatment of Lévy processes.

For each $d \in \mathbb{N}$ we consider a payoff function $\varphi_d\colon (0,\infty)^d \to \mathbb{R}$ and a Lévy process L^d with characteristic triplet $(\Sigma^d, \gamma^d, \nu_{\mathrm{L}}^d)$ satisfying $\nu_{\mathrm{L}}^d(\{y \in \mathbb{R}^d \,|\, \|y\| > R\}) = 0$ for some $R > 1$. We define the shifted drift vector $\tilde{\gamma}^d$ given by $\tilde{\gamma}_i^d = \gamma_i^d + \frac{1}{2}\Sigma_{i,i}^d + \int_{\mathbb{R}^d}(e^{y_i} - 1 - y_i\mathbb{1}_{\{\|y\|\leq 1\}})\nu_{\mathrm{L}}^d(dy)$ for $i = 1,\dots,d$. We now consider the partial (integro-)differential equation

$$\begin{aligned} \partial_t u_d(t,s) &= \tfrac{1}{2}\textstyle\sum_{k,l=1}^d s_k s_l \Sigma_{k,l}^d \partial_{s_k}\partial_{s_l} u_d(t,s) + \sum_{i=1}^d s_i \tilde{\gamma}_i^d \partial_{s_i} u_d(t,s) \\ &\quad + \textstyle\int_{\mathbb{R}^d}\left[u_d(t,se^y) - u_d(t,s) - \sum_{i=1}^d (e^{y_i}-1)s_i\partial_{s_i}u_d(t,s)\right]\nu_{\mathrm{L}}^d(dy), \\ u_d(0,s) &= \varphi_d(s) \end{aligned} \tag{41}$$

for $s \in (0,\infty)^d, t > 0$, where we write $s\exp(x) = (s_1\exp(x_1),\dots,s_d\exp(x_d))$ for $s, x \in \mathbb{R}^d$. The (non-local) PDE (41) is the Kolmogorov PDE for the exponential Lévy model associated to L^d. By Sato (1999, Theorem 25.17) the exponential Lévy process $(\exp(L_t^d))_{t\geq 0}$ is a martingale if (and only if) $\tilde{\gamma}^d = 0$. In this case, $u_d(T,s)$ is the price at time 0 of an option with payoff φ_d at maturity T when price of the underlying at time 0 is s. Furthermore, if the jump-measure vanishes ($\nu_{\mathrm{L}}^d = 0$), then (41) is the Black-Scholes PDE.

We now provide sufficient conditions on the payoff functions and the characteristic triplets which guarantee that $u_d(T,\cdot)$ can be approximated by random neural networks without the curse of dimensionality. To achieve this, the weights of the random neural networks are generated as follows: let $\nu > 1$ and for each $d \in \mathbb{N}$ let $A_1^d, A_2^d, \dots$ by i.i.d. $t_\nu(0, \mathbb{1}_d)$-distributed $\mathbb{R}^d$-valued random vectors independent of the i.i.d. random variables $B_1, B_2, \dots$ which have a strictly positive Lebesgue-density π_{b} of at most polynomial decay (see (26)). For $N \in \mathbb{N}$ we write $A^{d,N} = (A_1^d, \dots, A_N^d)$ and $B^N = (B_1, \dots, B_N)$.

Theorem 13 complements the results in Grohs et al. (2023), Gonon and Schwab (2021).

Theorem 13 *Let $p \geq 0$, $c, C, M, T > 0$. For each $d \in \mathbb{N}$ assume the payoff function satisfies $\varphi_d \circ \exp \in L^1(\mathbb{R}^d)$ and $\|\varphi_d \circ \exp\|_{L^1(\mathbb{R}^d)} \leq cd^p$, the characteristic triplet $(\Sigma^d, \gamma^d, \nu_{\mathrm{L}}^d)$ of the Lévy process L^d satisfies for all $\xi \in \mathbb{R}^d$*

$$\frac{1}{2}\xi \cdot \Sigma^d \xi \geq C\|\xi\|^2, \tag{42}$$

assume $CT > \frac{1}{2^{3/2}\pi}$ and suppose $u_d \in C^{1,2}((0,T] \times (0,\infty)^d) \cap C([0,T] \times (0,\infty)^d)$ is an at most polynomially growing solution to the PDE (41). Then there exist constants $C_0, \mathfrak{p} > 0$ such that for any $d, N \in \mathbb{N}$ there exists an $\mathbb{R}^N$-valued, $\sigma(A^{d,N}, B^N)$-measurable random vector $W^{d,N}$ such that the random neural network

$$\bar{H}_{d,N}(x) := H_{W^{d,N}}^{A^{d,N},B^N}(x) = \sum_{i=1}^{N} W_i^{d,N} \varrho(A_i^d \cdot x + B_i), \quad x \in \mathbb{R}^d, \tag{43}$$

satisfies the approximation bound

$$\mathbb{E}\left[\sup_{x \in [-M,M]^d} |\bar{H}_{d,N}(x) - u_d(T, \exp(x))|\right] \leq \frac{C_0 d^{\mathfrak{p}}}{\sqrt{N}}. \tag{44}$$

Proof Let $d, N \in \mathbb{N}$, $\Phi(x) = \varphi_d(\exp(x))$ and $H(x) = u_d(T, \exp(x))$ for $x \in \mathbb{R}^d$. Then Proposition 16 below shows that $H(x) = \mathbb{E}[\varphi_d(\exp(x + L_T^d))] = \mathbb{E}[\Phi(x + L_T^d)]$.

Furthermore, by the Lévy-Khintchine representation (see for instance Sato (1999, Theorem 8.1) or Applebaum (2009, Theorem 1.2.14 and Theorem 1.3.3)) we have $\mathbb{E}[e^{i\xi \cdot L_T^d}] = \exp(T\eta(\xi))$ with

$$\eta(\xi) = i\xi \cdot \gamma^d - \frac{1}{2}\xi \cdot \Sigma^d \xi + \int_{\mathbb{R}^d \setminus \{0\}} \left[e^{i\xi \cdot y} - 1 - i\xi \cdot y \mathbb{1}_{\{\|y\| \leq 1\}}\right] \nu_{\mathrm{L}}^d(dy), \quad \xi \in \mathbb{R}^d. \tag{45}$$

In particular,

$$\operatorname{Re}\eta(\xi) = -\frac{1}{2}\xi \cdot \Sigma^d \xi + \int_{\mathbb{R}^d \setminus \{0\}} [\cos(\xi \cdot y) - 1] \, \nu_{\mathrm{L}}^d(dy) \leq -\frac{1}{2}\xi \cdot \Sigma^d \xi,$$

since the integrability property $\int_{\mathbb{R}^d}(\|y\|^2 \wedge 1)\nu_{\mathrm{L}}^d(dy) < \infty$ guarantees that $y \mapsto \cos(\xi \cdot y) - 1$ and $y \mapsto \sin(\xi \cdot y) - \xi \cdot y \mathbb{1}_{\{\|y\| \leq 1\}}$ are indeed ν_{L}^d-integrable for any $\xi \in \mathbb{R}^d$. This and (42) show that for all $\xi \in \mathbb{R}^d$

$$|\mathbb{E}[e^{i\xi \cdot L_T^d}]| = e^{T\operatorname{Re}\eta(\xi)} \leq \exp(-CT\|\xi\|^2). \tag{46}$$

Theorem 7 hence shows that there exist $C_{\mathrm{app}} > 0$, $k \in \mathbb{N}$ and an $\mathbb{R}^N$-valued, $\sigma(A^{d,N}, B^N)$-measurable random vector $W^{d,N}$ such that the random neural network $\bar{H}_{d,N} = H_{W^{d,N}}^{A^{d,N},B^N}$ satisfies

$$\mathbb{E}\left[\sup_{x \in [-M,M]^d} |\bar{H}_{d,N}(x) - H(x)|\right] \leq \frac{C_{\mathrm{app}} \|\Phi\|_{L^1(\mathbb{R}^d)} (\nu + d)^{k+3}}{\sqrt{N}}. \tag{47}$$

Thus, we obtain

$$\mathbb{E}\left[\sup_{x\in[-M,M]^d}|\bar{H}_{d,N}(x)-u_d(T,\exp(x))|\right]\leq\frac{C_{\mathrm{app}}cd^p(\nu+1)^{k+3}d^{k+3}}{\sqrt{N}}=\frac{C_0d^{\mathfrak{p}}}{\sqrt{N}}$$

with $C_0=(\nu+1)^{k+3}C_{\mathrm{app}}c$ and $\mathfrak{p}=p+k+3$. This proves (44) and the statement, since C_{app} in Theorem 7 does not depend on d or N and hence the constants $C_0,\mathfrak{p}$ are the same for all $d,N\in\mathbb{N}$. ■

Remark 14 *Theorem 13 also holds if we directly assume $u_d(T,\exp(x))=\mathbb{E}[\varphi_d(\exp(x+L_T^d))]$ instead of considering the PDE (41). For instance in the context of mathematical finance many quantities of interest (such as option prices or "greeks") are defined in terms of such expectations. In particular, in this situation the hypothesis $\varphi_d\in C((0,\infty)^d,\mathbb{R})$ is not required (in Theorem 13 this hypothesis is implicit in the assumption $u_d\in C^{1,2}((0,T]\times(0,\infty)^d)\cap C([0,T]\times(0,\infty)^d)$).*

The integrability hypothesis $\varphi_d\circ\exp\in L^1(\mathbb{R}^d)$ is more restrictive, but currently it can not be avoided in the proof of Theorem 7. The hypothesis is satisfied e.g. for butterfly or binary options. More general payoffs can be incorporated by truncation (which is often possible without affecting the price significantly) or potentially by employing Fourier representations as in Carr and Madan (1999) instead of (30).

Remark 15 *The assumption $\nu_{\mathrm{L}}^d(\{y\in\mathbb{R}^d\,|\,\|y\|>R\})=0$ for some $R>1$ is only required to obtain a "Feynman-Kac representation" from the results of Barles et al. (1997) (see Proposition 16 below). This assumption on ν_{L}^d can be weakened to $\int_{\{\|y\|>1\}}e^{y_i}\nu_{\mathrm{L}}^d(dy)<\infty$ for $i=1,\dots,d$ for instance in the situation of Remark 14 when we directly assume a stochastic representation for u_d.*

Alternatively, instead of assuming $\nu_{\mathrm{L}}^d(\{y\in\mathbb{R}^d\,|\,\|y\|>R\})=0$ for some $R>1$ we could impose that ν_{L}^d is a finite measure and (42) holds. Then we may apply Pham (1998, Proposition 5.3) instead of Barles et al. (1997) in the proof of Proposition 16 below and also obtain the representation $u_d(t,s)=\mathbb{E}[\varphi_d(s\exp(L_t^d))]$.

The proof of Theorem 13 employs the "Feynman-Kac representation" from Proposition 16 below. Proposition 16 is essentially a consequence of the results from Barles et al. (1997). For the readers' convenience we provide a proof of Proposition 16 and make explicit how it can be obtained from Barles et al. (1997). Related results and further references can be found, for instance, in Pham (1998), Cont and Voltchkova (2005), Cont and Voltchkova (2006, Proposition 3.3), Glau (2016).

Proposition 16 *Suppose $u_d\in C^{1,2}((0,T]\times(0,\infty)^d)\cap C([0,T]\times(0,\infty)^d)$ is an at most polynomially growing solution to the PDE (41) and φ_d is bounded. Then for all $(t,s)\in[0,T]\times(0,\infty)^d$ it holds that $u_d(t,s)=\mathbb{E}[\varphi_d(s\exp(L_t^d))]$.*

Proof Let $\Phi_d(x) = \varphi_d(\exp(x))$ and $v_d(t,x) = u_d(T-t, \exp(x))$. Firstly, the assumptions on u_d imply that $v_d \in C^{1,2}([0,T)\times\mathbb{R}^d) \cap C([0,T]\times\mathbb{R}^d)$ and a straightforward calculation shows that v_d satisfies the (non-local) PDE

$$\begin{aligned} -\partial_t v_d(t,x) &= \tfrac{1}{2}\sum_{k,l=1}^d \Sigma^d_{k,l}\partial_{x_k}\partial_{x_l} v_d(t,x) + \sum_{i=1}^d \left(\gamma_i^d + \textstyle\int_{\mathbb{R}^d} y_i \mathbb{1}_{\{\|y\|>1\}}\nu_{\mathrm{L}}^d(dy)\right)\partial_{x_i} v_d(t,x) \\ &\quad + \textstyle\int_{\mathbb{R}^d}\left[v_d(t,x+y) - v_d(t,x) - \sum_{i=1}^d y_i\partial_{x_i} v_d(t,x)\right]\nu_{\mathrm{L}}^d(dy), \\ v_d(T,x) &= \Phi_d(x) \end{aligned} \tag{48}$$

for $x\in\mathbb{R}^d, t\in[0,T)$. Set $\hat{\gamma}^d = (\gamma^d + \int_{\mathbb{R}^d} y\mathbb{1}_{\{\|y\|>1\}}\nu_{\mathrm{L}}^d(dy))$ and for $\phi\in C^2(\mathbb{R}^d)$ write

$$\begin{aligned} \mathcal{A}\phi(x) &= \frac{1}{2}\mathrm{Trace}(\Sigma^d D_x^2\phi(x)) + [D_x\phi(x)]\hat{\gamma}^d \\ \mathcal{K}\phi(x) &= \int_{\mathbb{R}^d}(\phi(x+y) - \phi(x) - [D_x\phi(x)]y)\nu_{\mathrm{L}}^d(dy). \end{aligned} \tag{49}$$

Now if $\phi\in C^2([0,T]\times\mathbb{R}^d)$ and $(t_0,x_0)\in[0,T)\times\mathbb{R}^d$ is a global maximum point of $v_d-\phi$, then $D_{(t,x)}(v_d-\phi)(t_0,x_0)=0$ and $D_x^2(v_d-\phi)(t_0,x_0)\le 0$. Thus, (48) implies

$$\begin{aligned} &-\partial_t\phi(t_0,x_0) - \mathcal{A}\phi(t_0,x_0) - \mathcal{K}\phi(t_0,x_0) \\ &\quad = \mathcal{A}(v_d-\phi)(t_0,x_0) + \mathcal{K}(v_d-\phi)(t_0,x_0) \\ &\quad = \frac{1}{2}\mathrm{Trace}(\sqrt{\Sigma^d}D_x^2(v_d-\phi)(t_0,x_0)\sqrt{\Sigma^d}) + \int_{\mathbb{R}^d}(v_d-\phi)(t_0,x_0+y) - (v_d-\phi)(t_0,x_0)\nu_{\mathrm{L}}^d(dy) \\ &\quad \le 0. \end{aligned} \tag{50}$$

This and Barles et al. (1997, Lemma 3.3) show that v_d is a viscosity subsolution of (48) in the sense of Barles et al. (1997). Similarly, one argues that v_d is also a viscosity supersolution to (48). Barles et al. (1997, Theorem 3.5) hence shows that for all $(t,x)\in[0,T]\times\mathbb{R}^d$ we have $v_d(t,x) = \mathbb{E}[\Phi_d(X_T^{t,x})]$ (see also the proof of Gonon and Schwab (2023, Corollary 5.4)) where $(X_r^{t,x})_{r\ge t}$ is the unique solution to $X_t^{t,x} = x$,

$$\begin{aligned} dX_r^{t,x} &= \hat{\gamma}^d dr + \sqrt{\Sigma^d}W_r^d + \int_{\mathbb{R}^d\setminus\{0\}} z\tilde{N}^d(dt,dz) \\ &= \gamma^d dr + \sqrt{\Sigma^d}W_r^d + \int_{\mathbb{R}^d\setminus\{0\}} z\mathbb{1}_{\{\|z\|\le 1\}}\tilde{N}^d(dr,dz) + \int_{\mathbb{R}^d\setminus\{0\}} z\mathbb{1}_{\{\|z\|>1\}}N^d(dr,dz) \end{aligned}$$

where N^d is a Poisson random measure on $\mathbb{R}_+\times(\mathbb{R}^d\setminus\{0\})$ with intensity ν_{L}^d, W^d is an independent d-dimensional standard Brownian motion and $\tilde{N}^d(dt,dz) = N^d(dt,dz) - dt\nu_{\mathrm{L}}^d(dz)$. Note that the assumption $\nu_{\mathrm{L}}^d(\{y\in\mathbb{R}^d \mid \|y\|>R\}) = 0$ for some $R>1$ guarantees that the function β in Barles et al. (1997, Theorem 3.5) can be chosen so that it satisfies the required boundedness hypothesis. Hence, by the Lévy-Itô-decomposition (see for instance Sato 1999, Theorem 19.2 or Applebaum 2009, Theorem 2.4.16) we obtain that $X_T^{t,x}$ has the same distribution as $x + L_{T-t}^d$. Thus, we have proved the representation $v_d(t,x) = \mathbb{E}[\Phi_d(x+L_{T-t}^d)]$ and therefore for all $x\in\mathbb{R}^d$, with $s=\exp(x)$,

$$u_d(t,s) = v_d(T-t,x) = \mathbb{E}[\varphi_d(\exp(x+L_t^d))] = \mathbb{E}[\varphi_d(s\exp(L_t^d))].$$

■

4. Learning by Random Neural Networks

In this section we use random neural networks $H_W^{A,B}$ to learn functions of the type considered in Section 3.1. In Section 4.1 we formulate the considered learning problem. In Sections 4.2, 4.3, 4.4 we then provide bounds on the prediction error that arises when W is learnt by means of regression, constrained regression and stochastic gradient descent, respectively. In Sections 4.5 we will then apply these results to obtain prediction error bounds for random neural networks applied to learning option prices in certain non-degenerate models.

4.1 Formulation of the Learning Problem

Let $n \in \mathbb{N}$ and suppose that we are given i.i.d. $\mathbb{R}^d \times \mathbb{R}$-valued random variables $(X_1, Y_1), \dots, (X_n, Y_n)$ (the data) which are independent of (A, B). Let $H\colon \mathbb{R}^d \to \mathbb{R}$ be the target function (which we will assume to be of the form specified in Section 3.1) and suppose that

$$H(x) = \mathbb{E}[Y_1 | X_1 = x], \tag{51}$$

for $(\mathbb{P} \circ (X_1)^{-1})$-a.e. $x \in \mathbb{R}^d$, that is, H is the regression function. This encompasses two important situations:

- *Learning H from noisy observations*: We observe the unknown function H (the solution to a PDE or market prices of options) at n data points up to some additive noise. Thus, in this situation we suppose $Y_i = H(X_i) + \varepsilon_i$, $i = 1, \dots, n$, for $\varepsilon_1, \dots, \varepsilon_n$ i.i.d. random variables which are independent of $(X_1, \dots, X_n)$ and satisfy $\mathbb{E}[\varepsilon_1] = 0$.
- *Solving PDEs by learning*: Solving linear Kolmogorov PDEs with affine coefficients has been formulated as a learning problem in Berner et al. (2020). The setting considered here also covers this type of learning problem.

The target function H is considered unknown and is to be learnt from the data $D_n = ((X_1, Y_1), \dots, (X_n, Y_n))$ using random neural networks. To do this, we recall that $H(X_1) = \mathbb{E}[Y_1|X_1]$ minimizes

$$\mathcal{R}(f) = \mathbb{E}[(f(X_1) - Y_1)^2] \tag{52}$$

among all measurable functions $f\colon \mathbb{R}^d \to \mathbb{R}$. Thus, to learn $H(x) = \mathbb{E}[Y_1|X_1 = x]$ from the data one aims at finding a minimizer of

$$\mathcal{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2. \tag{53}$$

$\mathcal{R}_n(f)$ is the empirical version of (52). In the situation considered here we know from Section 3 that H can be approximated well by random neural networks and so we learn H by minimizing $\mathcal{R}_n(\cdot)$ only over this class of functions, i.e. by minimizing $\mathcal{R}_n(H_W^{A,B})$ over neural networks $H_W^{A,B}$ with random weights (A, B) and trainable W (see Section 2). This leads to the optimization problem

$$\widehat{W} = \arg\min_{W \in \mathcal{W}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (H_W^{A,B}(X_i) - Y_i)^2 \right\} \tag{54}$$

for a suitable set $\mathcal{W}$ of $\mathbb{R}^N$-valued, $\sigma(A, B, D_n)$-measurable random vectors. The measurability requirement incorporates the fact that A, B are generated randomly and then fixed and hence the trainable weights may depend on A, B.

Having solved (54), the learning algorithm then returns the (random) function

$$H_{\widehat{W}}^{A,B}(x) = \sum_{i=1}^{N} \widehat{W}_i \varrho(A_i \cdot x + B_i), \quad x \in \mathbb{R}^d$$

as our approximation for H. To evaluate the learning performance of the random features regression algorithm we need to bound the (squared) learning error (or prediction error)

$$\mathbb{E}[|H(\bar{X}) - H_{\widehat{W}}^{A,B}(\bar{X})|^2], \tag{55}$$

where $(\bar{X}, \bar{Y})$ has the same distribution as (X_1, Y_1) and is independent of (A, B, D_n).

4.2 Regression

Consider first the case $\mathcal{W} = \{W \colon \Omega \to \mathbb{R}^N \,|\, W \text{ is } \sigma(A, B, D_n)\text{-measurable}\}$. In this case computing (54) amounts to a simple least squares optimization. Hence $\widehat{W}$ can be calculated explicitly by solving

$$(\mathbf{X}^\top \mathbf{X})\widehat{W} = \mathbf{X}^\top \mathbf{Y} \tag{56}$$

where $\mathbf{X}$ is the $n \times N$-random matrix with entries $\mathbf{X}_{ij} = \varrho(A_j \cdot X_i + B_j)$ and $\mathbf{Y}$ is the n-dimensional random vector with $\mathbf{Y}_i = Y_i$ for $i = 1, \dots, n$, $j = 1, \dots, N$.

Thus, there is no additional "optimization error" component in this case and we can directly bound the prediction error (55) by combining the approximation error estimates from Section 3 with a result from Györfi et al. (2002).

The trained neural network $H_{\widehat{W}}^{A,B}$ will be capped at a level $L > 0$ by applying the truncation $T_L \colon \mathbb{R} \to \mathbb{R}$, $T_L(u) = \max(\min(u, L), -L)$.

Theorem 17 *Let $C > \frac{1}{2^{3/2}\pi}$ and let $\nu > 1$. Suppose $A_1 \sim t_\nu(0, \mathbb{1}_d)$ and B_1 has density π_b satisfying (26). Suppose $H \colon \mathbb{R}^d \to \mathbb{R}$ is of the form $H(x) = \mathbb{E}[\Phi(x + V)]$ with $\Phi \in L^1(\mathbb{R}^d)$ and V satisfying (25). Assume that $\|X_1\|_\infty \le M$, $\mathbb{P}$-a.s. Let $L > 0$ and assume $\sigma^2 = \sup_{x \in \mathbb{R}^d} \mathbb{E}[(Y_1 - H(X_1))^2 | X_1 = x] < \infty$ and $|H(x)| \le L$ for all $x \in \mathbb{R}^d$. Then there exist $k \in \mathbb{N}$ and $\tilde{C}_{app} > 0$ such that*

$$\begin{aligned} &\mathbb{E}[|H(\bar{X}) - T_L(H_{\widehat{W}}^{A,B}(\bar{X}))|^2]^{1/2} \\ &\le \tilde{C}_{app} \max(\sigma, L) \frac{(\log(n) + 1)^{1/2}\sqrt{N}}{\sqrt{n}} + \frac{\tilde{C}_{app} \|\Phi\|_{L^1(\mathbb{R}^d)} (\nu + d)^{k+3}}{\sqrt{N}}. \end{aligned} \tag{57}$$

The constant k only depends on π_b and the constant $\tilde{C}_{app}$ depends on ν, π_b, C, M, but it does not depend on d, n or N.

Remark 18 *Theorem 17 bounds the square-root of the prediction error by $\mathcal{O}(\frac{\log(n)^{1/2}\sqrt{N}}{\sqrt{n}} + \frac{1}{\sqrt{N}})$. This matches, up to constants, the error bound obtained in the seminal work Barron (1994) for general "Barron functions". In Barron (1994) all parameters of the network are*

trainable and the neural network estimator is defined via empirical risk minimization over a constrained parameter set. However, the optimization error, which arises when the neural network estimator is calculated based e.g. on the stochastic gradient descent algorithm, is not addressed in Barron (1994). In contrast, in our situation the class of considered functions is smaller, but the neural network estimator can be directly calculated by solving the linear system (56)*. Hence, the bound in Theorem 17 captures the full training error.*

Proof Firstly, for fixed $a \in (\mathbb{R}^d)^N$, $b \in \mathbb{R}^N$ we consider the function class $\mathcal{F}_{a,b} = \{H_W^{a,b} \,|\, W \in \mathbb{R}^N\}$, $\mathcal{F}_{a,b}(D_n) = \{H_W^{a,b} \,|\, W\colon \Omega \to \mathbb{R}^N \text{ is } \sigma(D_n)\text{-measurable}\}$ (in Györfi et al. (2002) the same symbol is used for these two sets) and let $\hat{f}_{a,b} = \arg\min_{f \in \mathcal{F}_{a,b}(D_n)} \mathcal{R}_n(f)$. Then $\mathcal{F}_{a,b}$ is an N-dimensional vector space and hence Györfi et al. (2002, Theorem 11.3) implies that

$$\begin{aligned} &\mathbb{E}\left[\int_{\mathbb{R}^d} |T_L(\hat{f}_{a,b}(x)) - H(x)|^2 \mu_X(dx)\right] \\ &\quad \leq c \max(\sigma^2, L^2) \frac{(\log(n)+1)N}{n} + 8 \inf_{f \in \mathcal{F}_{a,b}} \int_{\mathbb{R}^d} |f(x) - H(x)|^2 \mu_X(dx), \end{aligned} \tag{58}$$

where μ_X is the law of X_1 under $\mathbb{P}$ and $c = 8 + 2304[\log(9) + 4\log(12e) + 1]$.

For any $a \in (\mathbb{R}^d)^N$, $b \in \mathbb{R}^N$ the minimization problem for $\hat{f}_{a,b}$ can be solved explicitly and we obtain $\hat{f}_{a,b} = H^{a,b}_{\hat{w}_{a,b}}$, where $\hat{w}_{a,b}$ is a solution to the linear system (56) with A, B fixed to a, b. A solution always exists (see for instance Stoer and Bulirsch 2002, Chapter 4.8.1) and, e.g. by choosing the solution given in terms of the pseudo-inverse matrix as $\widehat{W} = (\mathbf{X}^\top \mathbf{X})^\dagger \mathbf{X}^\top \mathbf{Y}$, it is possible to write $\widehat{W} = F(A, B, D_n)$ for a measurable function $F\colon (\mathbb{R}^d)^N \times \mathbb{R}^N \times (\mathbb{R}^d \times \mathbb{R})^n \to \mathbb{R}^N$ and select $\hat{w}_{a,b}$ in such a way that $\hat{w}_{a,b} = F(a, b, D_n)$.

Using independence we thus obtain from (58)

$$\begin{aligned} &\mathbb{E}\left[|T_L(H^{A,B}_{\widehat{W}}(\bar{X})) - H(\bar{X})|^2 | A, B\right] = \mathbb{E}[|T_L(H^{a,b}_{\hat{w}_{a,b}}(\bar{X})) - H(\bar{X})|^2]\Big|_{(a,b)=(A,B)} \\ &\quad \leq c \max(\sigma^2, L^2) \frac{(\log(n)+1)N}{n} + 8 \left(\inf_{W \in \mathbb{R}^N} \mathbb{E}[|H_W^{a,b}(\bar{X}) - H(\bar{X})|^2]\right)\Big|_{(a,b)=(A,B)} \\ &\quad \leq c \max(\sigma^2, L^2) \frac{(\log(n)+1)N}{n} + 8\mathbb{E}[|H^{A,B}_{W^*}(\bar{X}) - H(\bar{X})|^2 | A, B], \end{aligned} \tag{59}$$

where W^* denotes the random vector from Theorem 7. We may therefore take expectations in (59), use $\|\bar{X}\|_\infty \leq M$ and insert the bound from Theorem 7 (c.f. also Remark 8) to deduce (57) with $\tilde{C}_{\text{app}} = \max(\sqrt{c}, \sqrt{8} C_{\text{app}})$. ∎

4.3 Constrained Regression

In the next result we consider a constrained regression estimator, i.e., $\widehat{W}$ in (54) is calculated with a smaller set of potential weights $\mathcal{W}$. This leads to a different bound than in Theorem 17, but for instance for $N = \sqrt{n}$ the same rate is achieved.

Set $\mathcal{W}_\lambda = \{W\colon \Omega \to \mathbb{R}^N \,|\, W \text{ is } \sigma(A, B, D_n)\text{-measurable}, \|W\| \leq \lambda \ \mathbb{P}\text{-a.s.}\}$. Computing (54) now corresponds to a constrained regression problem

$$\widehat{W}_\lambda = \arg\min_{W \in \mathcal{W}_\lambda} \left\{ \frac{1}{n} \sum_{i=1}^n (H_W^{A,B}(X_i) - Y_i)^2 \right\}. \tag{60}$$

The solution to (60) is given explicitly as follows: $\widehat{W}_\lambda$ coincides with the solution $\widehat{W}$ to the unconstrained problem (56) with minimal norm in case $\widehat{W}$ satisfies $\|\widehat{W}\| \leq \lambda$. Otherwise $\widehat{W}_\lambda$ is given explicitly as

$$\widehat{W}_\lambda = (\mathbf{X}^\top \mathbf{X} + \mathbb{1}\Lambda)^{-1} \mathbf{X}^\top \mathbf{Y} \tag{61}$$

with Λ a non-negative $\sigma(A, B, D_n)$-measurable random variable[2] such that $\|\widehat{W}_\lambda\| = \lambda$. The two cases can be summarized by setting $\Lambda = 0$ in the first case and interpreting the inverse in (61) as a pseudo-inverse, then $\widehat{W}_\lambda$ is given by (61) in both cases.

We now provide a bound on the prediction error for random neural networks with parameters learned according to (60).

Theorem 19 *Let $C > \frac{1}{2^{3/2}\pi}$ and let $\nu > 2$. Suppose $A_1 \sim t_\nu(0, \mathbb{1}_d)$ and B_1 has density π_b satisfying (26). Suppose $H \colon \mathbb{R}^d \to \mathbb{R}$ is of the form $H(x) = \mathbb{E}[\Phi(x + V)]$ with $\Phi \in L^1(\mathbb{R}^d)$ and V satisfying (25). Assume that $\|X_1\|_\infty \leq M$, $\mathbb{P}$-a.s. and $\mathbb{E}[|Y_1|^4] < \infty$. Let $k \in \mathbb{N}$ and $C_{app}, C_{wgt} > 0$ be as in Theorem 7. Let $\lambda > 0$ satisfy $\frac{C_{wgt}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu+d)^{2k+\frac{1}{2}}}{\sqrt{N}} \leq \lambda \leq \frac{C_{lam} d^p}{\sqrt{N}}$ for some $p \geq 0$, $C_{lam} > 0$ not depending on n, N, d. Then there exists $C_{est} > 0$ such that*

$$\mathbb{E}[|H(\bar{X}) - H^{A,B}_{\widehat{W}_\lambda}(\bar{X})|^2]^{1/2} \leq \frac{C_{app}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu + d)^{k+3}}{\sqrt{N}} + \frac{C_{est} d^{p+1}}{n^{\frac{1}{4}}}. \tag{62}$$

The constant C_{est} depends on $\nu, \pi_b, C_{lam}, M, \mathbb{E}[Y_1^4]$, but it does not depend on d, n or N.

Remark 20 *Theorem 19 shows that the prediction error is of order $\mathcal{O}(\frac{1}{N} + \frac{1}{\sqrt{n}})$. Thus, the error bound decays more quickly than the bound $\mathcal{O}(\frac{1}{\sqrt{N}} + \frac{1}{\sqrt{n}})$ that was obtained in the seminal work Rahimi and Recht (2009), where high-probability bounds were obtained for random neural networks trained by constrained regression in a classification setting ($\mathbb{P}(Y_i \in \{1, -1\}) = 1$). The reason for this faster rate is that we use the mean-square loss here. This allows to write $|\mathcal{R}(H) - \mathcal{R}(\tilde{H})| = \mathbb{E}[|H(\bar{X}) - \tilde{H}(\bar{X})|^2]$ due to (51). For L-Lipschitz loss functions the bound $\mathcal{R}(H) - \mathcal{R}(\tilde{H}) \leq L\mathbb{E}[|H(\bar{X}) - H^{A,B}_{\widehat{W}_\lambda}(\bar{X})|^2]^{1/2}$ can be deduced (see Rahimi and Recht 2009, Lemma 2), which leads to an approximation error of order $1/\sqrt{N}$ instead of $1/N$.*

Thus, we are concerned here with a slightly different setting, but our proof of the "estimation error" (or generalization error) component is based on similar arguments as the proof in Rahimi and Recht (2009).

Proof Firstly, (51) and independence imply

$$\begin{aligned}
\mathbb{E}[H(\bar{X}) H^{A,B}_{\widehat{W}_\lambda}(\bar{X})] &= \mathbb{E}[\mathbb{E}[\mathbb{E}[\bar{Y}|\bar{X}] H^{a,b}_w(\bar{X})]\Big|_{(a,b,w)=(A,B,\widehat{W}_\lambda)}] \\
&= \mathbb{E}[\mathbb{E}[\bar{Y} H^{a,b}_w(\bar{X})]\Big|_{(a,b,w)=(A,B,\widehat{W}_\lambda)}] \\
&= \mathbb{E}[\bar{Y} H^{A,B}_{\widehat{W}_\lambda}(\bar{X})]
\end{aligned} \tag{63}$$

2. This means that once the data and the random weights have been sampled/observed (i.e. conditionally on these) Λ is just a constant.

and analogously $\mathbb{E}[H(\bar{X})H_W^{A,B}(\bar{X})] = \mathbb{E}[\bar{Y}H_W^{A,B}(\bar{X})]$ for any $W \in \mathcal{W}_\lambda$. Thus, we calculate

$$\begin{aligned}
&\mathbb{E}[|H(\bar{X}) - H_{\widehat{W}_\lambda}^{A,B}(\bar{X})|^2] \\
&\quad = \mathbb{E}[|H(\bar{X}) - H_W^{A,B}(\bar{X})|^2] + \mathbb{E}[|H_{\widehat{W}_\lambda}^{A,B}(\bar{X}) - \bar{Y}|^2] - \mathbb{E}[|H_W^{A,B}(\bar{X}) - \bar{Y}|^2] \\
&\quad = \mathbb{E}[|H(\bar{X}) - H_W^{A,B}(\bar{X})|^2] + \mathbb{E}[\mathcal{R}(H_{\widehat{W}_\lambda}^{A,B}) - \mathcal{R}(H_W^{A,B})] \\
&\quad \leq \mathbb{E}[|H(\bar{X}) - H_W^{A,B}(\bar{X})|^2] + \mathbb{E}[\mathcal{R}(H_{\widehat{W}_\lambda}^{A,B}) - \mathcal{R}_n(H_{\widehat{W}_\lambda}^{A,B}) + \mathcal{R}_n(H_W^{A,B}) - \mathcal{R}(H_W^{A,B})],
\end{aligned} \tag{64}$$

where we used (60) and $W \in \mathcal{W}_\lambda$ in the last step.

Consider the first term in the right hand side of (64). Theorem 7 (c.f. also Remark 8) guarantees that there exists an $\mathbb{R}^N$-valued, $\sigma(A,B)$-measurable random vector W^* such that

$$\mathbb{E}[|H(\bar{X}) - H_{W^*}^{A,B}(\bar{X})|^2]^{1/2} \leq \frac{C_{\text{app}}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu+d)^{k+3}}{\sqrt{N}}, \tag{65}$$

where we used that $\|\bar{X}\|_{L^\infty(\mathbb{P})} \leq M$. Furthermore, (28) shows that $\mathbb{P}$-a.s. the weight vector satisfies $\|W^*\| \leq \sqrt{N}\max_{i=1}^N \|W_i^*\|_{L^\infty(\mathbb{P})} \leq \frac{C_{\text{wgt}}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu+d)^{2k+\frac{1}{2}}}{\sqrt{N}} \leq \lambda$. Hence, it follows that $W^* \in \mathcal{W}_\lambda$ and so the decomposition (64) can be applied with $W = W^*$.

For the second term in the right hand side of (64) we let $\widehat{W}_\lambda^{a,b}$ denote the solution to (60) for (A,B) fixed to (a,b). The random variable Λ can be written as $\Lambda = F(A,B,D_n)$ for a measurable function $F\colon (\mathbb{R}^d)^N \times \mathbb{R}^N \times (\mathbb{R}^d \times \mathbb{R})^n \to [0,\infty)$ (in fact, $F(a,b,d_n) = \inf\{t \geq 0 \,|\, f_{a,b,d_n}(t) \leq \lambda\}$ for the strictly decreasing function $f_{a,b,d_n}(t) = \|(\mathbf{X}_{a,b,d_n}^\top \mathbf{X}_{a,b,d_n} + \mathbb{1}t)^{-1}\mathbf{X}_{a,b,d_n}^\top \mathbf{Y}_{a,b,d_n}\|$, where $\mathbf{X}_{a,b,d_n}, \mathbf{Y}_{a,b,d_n}$ are $\mathbf{X},\mathbf{Y}$ with (A,B,D_n) fixed to (a,b,d_n)). Then from the formula (61) it is clear that $\widehat{W}_\lambda = G(A,B,D_n)$ for a measurable function G and $\widehat{W}_\lambda^{a,b} = G(a,b,D_n)$. Furthermore, we write $(a,b) \mapsto W^{a,b}$ for the measurable function with $W^{A,B} = W$ (which exists, since W is $\sigma(A,B)$-measurable) and $\mathcal{W}_\lambda^0 = \{w \in \mathbb{R}^N \,|\, \|w\| \leq \lambda\}$. Then by independence

$$\begin{aligned}
&\mathbb{E}[\mathcal{R}(H_{\widehat{W}_\lambda}^{A,B}) - \mathcal{R}_n(H_{\widehat{W}_\lambda}^{A,B}) + \mathcal{R}_n(H_W^{A,B}) - \mathcal{R}(H_W^{A,B})] \\
&= \mathbb{E}[\mathbb{E}[\mathcal{R}(H_{\widehat{W}_\lambda^{a,b}}^{a,b}) - \mathcal{R}_n(H_{\widehat{W}_\lambda^{a,b}}^{a,b}) + \mathcal{R}_n(H_{W^{a,b}}^{a,b}) - \mathcal{R}(H_{W^{a,b}}^{a,b})]\Big|_{(a,b)=(A,B)}] \\
&\leq 2\mathbb{E}\left[\mathbb{E}\left[\sup_{w\in\mathcal{W}_\lambda^0}\left|\mathcal{R}(H_w^{a,b}) - \mathcal{R}_n(H_w^{a,b})\right|\right]\Big|_{(a,b)=(A,B)}\right].
\end{aligned} \tag{66}$$

We now fix (a,b), consider for $i = 1,\dots,n$, $w \in \mathcal{W}_\lambda^0$ the random variables $U_{w,i}^{a,b} = (H_w^{a,b}(X_i) - Y_i)^2$ and let $\varepsilon_1,\dots,\varepsilon_n$ denote i.i.d. Rademacher random variables independent of all other random variables. Employing symmetrization (see for instance Boucheron et al. 2013, Lemma 11.4) we obtain

$$\mathbb{E}\left[\sup_{w\in\mathcal{W}_\lambda^0}\left|\mathcal{R}(H_w^{a,b}) - \mathcal{R}_n(H_w^{a,b})\right|\right] \leq 2\mathbb{E}\left[\sup_{w\in\mathcal{W}_\lambda^0}\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i U_{w,i}^{a,b}\right|\right]. \tag{67}$$

In the next step we denote by $\mathbf{X}^i$ the vector with components $\mathbf{X}^i_j = \varrho(a_j \cdot X_i + b_j)$, $j = 1, \dots, N$ and rewrite $H^{a,b}_w(X_i) = w \cdot \mathbf{X}^i$. Then we use the triangle inequality, Jensen's inequality and independence to estimate

$$\begin{aligned}
\mathbb{E}\left[\sup_{w\in\mathcal{W}^0_\lambda}\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i U^{a,b}_{w,i}\right|\right] &\leq \mathbb{E}\left[\sup_{w\in\mathcal{W}^0_\lambda}\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i H^{a,b}_w(X_i)^2\right|\right] + \mathbb{E}\left[\sup_{w\in\mathcal{W}^0_\lambda}\left|\frac{2}{n}\sum_{i=1}^n \varepsilon_i H^{a,b}_w(X_i) Y_i\right|\right] \\
&\quad + \frac{1}{n}\mathbb{E}\left[\left|\sum_{i=1}^n \varepsilon_i Y_i^2\right|\right] \\
&\leq \mathbb{E}\left[\sup_{w\in\mathcal{W}^0_\lambda}\left|w^\top\left(\frac{1}{n}\sum_{i=1}^n \varepsilon_i \mathbf{X}^i[\mathbf{X}^i]^\top\right)w\right|\right] \\
&\quad + \mathbb{E}\left[\sup_{w\in\mathcal{W}^0_\lambda}\left|\frac{2}{n}w^\top\sum_{i=1}^n \varepsilon_i \mathbf{X}^i Y_i\right|\right] + \frac{1}{n}\mathbb{E}\left[\left|\sum_{i=1}^n \varepsilon_i Y_i^2\right|^2\right]^{1/2} \\
&\leq \frac{\lambda^2}{n}\mathbb{E}\left[\left\|\sum_{i=1}^n \varepsilon_i \mathbf{X}^i[\mathbf{X}^i]^\top\right\|_F^2\right]^{1/2} + \frac{2\lambda}{n}\mathbb{E}\left[\left\|\sum_{i=1}^n \varepsilon_i \mathbf{X}^i Y_i\right\|^2\right]^{1/2} \\
&\quad + \frac{1}{n}\left(\sum_{i=1}^n \mathbb{E}[Y_i^4]\right)^{1/2},
\end{aligned} \tag{68}$$

where $\|\cdot\|_F$ is the Frobenius norm on $\mathbb{R}^{N\times N}$. Denoting by $\langle\cdot,\cdot\rangle_F$ the Frobenius (matrix) inner product on $\mathbb{R}^{N\times N}$ and using independence and $\mathbb{E}[\varepsilon_i\varepsilon_j] = \delta_{ij}$ we obtain

$$\mathbb{E}\left[\left\|\sum_{i=1}^n \varepsilon_i \mathbf{X}^i[\mathbf{X}^i]^\top\right\|_F^2\right] = \mathbb{E}\left[\sum_{i,j=1}^n \varepsilon_i\varepsilon_j\langle \mathbf{X}^i[\mathbf{X}^i]^\top, \mathbf{X}^j[\mathbf{X}^j]^\top\rangle\right] = n\mathbb{E}\left[\left\|\mathbf{X}^1[\mathbf{X}^1]^\top\right\|_F^2\right].$$

Employing an analogous argument for the second term in the right hand side of (68) (now with the standard inner product on $\mathbb{R}^N$) yields

$$\begin{aligned}
\mathbb{E}\left[\sup_{w\in\mathcal{W}^0_\lambda}\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i U^{a,b}_{w,i}\right|\right] &\leq \frac{\lambda^2}{\sqrt{n}}\mathbb{E}\left[\left\|\mathbf{X}^1[\mathbf{X}^1]^\top\right\|_F^2\right]^{1/2} + \frac{2\lambda}{\sqrt{n}}\mathbb{E}\left[\left\|\mathbf{X}^1 Y_1\right\|^2\right]^{1/2} \\
&\quad + \frac{1}{\sqrt{n}}\mathbb{E}[Y_1^4]^{1/2}.
\end{aligned} \tag{69}$$

Using $\left\|\mathbf{X}^1[\mathbf{X}^1]^\top\right\|_F^2 = \sum_{k,l=1}^N [\mathbf{X}^1_k]^2[\mathbf{X}^1_l]^2 = \|\mathbf{X}^1\|^4$ and inserting the bound (69) in (67) we obtain

$$\begin{aligned}
\mathbb{E}\left[\sup_{w\in\mathcal{W}^0_\lambda}\left|\mathcal{R}(H^{a,b}_w) - \mathcal{R}_n(H^{a,b}_w)\right|\right] &\leq \frac{2\lambda^2}{\sqrt{n}}\mathbb{E}\left[\left\|\mathbf{X}^1\right\|^4\right]^{1/2} + \frac{4\lambda}{\sqrt{n}}\mathbb{E}\left[\left\|\mathbf{X}^1\right\|^2 Y_1^2\right]^{1/2} \\
&\quad + \frac{2}{\sqrt{n}}\mathbb{E}[Y_1^4]^{1/2}.
\end{aligned} \tag{70}$$

Employing the bound

$$\left\|\mathbf{X}^1\right\|^2 = \sum_{j=1}^{N} [\varrho(a_j \cdot X_1 + b_j)]^2 \leq 2 \sum_{j=1}^{N} \|a_j\|^2 \|X_1\|^2 + |b_j|^2 \tag{71}$$

we estimate using the Minkowski integral inequality and the triangle inequality

$$\begin{aligned}\mathbb{E}\left[\left\|\mathbf{X}^1\right\|^4\right]^{1/2} &\leq \mathbb{E}\left[\left(2\sum_{j=1}^{N} \|a_j\|^2\|X_1\|^2 + |b_j|^2\right)^2\right]^{1/2} \leq 2\sum_{j=1}^{N} \mathbb{E}\left[\left(\|a_j\|^2\|X_1\|^2 + |b_j|^2\right)^2\right]^{1/2} \\ &\leq 2\sum_{j=1}^{N} \|a_j\|^2 \mathbb{E}[\|X_1\|^4]^{1/2} + |b_j|^2.\end{aligned}$$

The second term in the right hand side of (70) can be bounded similarly with (71). Inserting this and (70) in (66) yields

$$\begin{aligned}&\mathbb{E}[\mathcal{R}(H^{A,B}_{\widehat{W}_\lambda}) - \mathcal{R}_n(H^{A,B}_{\widehat{W}_\lambda}) + \mathcal{R}_n(H^{A,B}_{W}) - \mathcal{R}(H^{A,B}_{W})] \\ &\leq 2\mathbb{E}\left[\frac{4\lambda^2}{\sqrt{n}}\left(\sum_{j=1}^{N} \|A_j\|^2 \mathbb{E}[\|X_1\|^4]^{1/2} + |B_j|^2\right)\right] \\ &\quad + 2\mathbb{E}\left[\frac{2^{2+\frac{1}{2}}\lambda}{\sqrt{n}}\left(\sum_{j=1}^{N} \|A_j\|^2 \mathbb{E}[Y_1^2\|X_1\|^2] + |B_j|^2\mathbb{E}[Y_1^2]\right)^{1/2}\right] + \frac{4}{\sqrt{n}}\mathbb{E}[Y_1^4]^{1/2} \\ &\leq \frac{8\lambda^2 N}{\sqrt{n}}(\mathbb{E}[\|A_1\|^2]\mathbb{E}[\|X_1\|^4]^{1/2} + \mathbb{E}[|B_1|^2]) \\ &\quad + \frac{2^{3+\frac{1}{2}}\lambda\sqrt{N}}{\sqrt{n}}\left(\mathbb{E}[\|A_1\|^2]\mathbb{E}[Y_1^2\|X_1\|^2] + \mathbb{E}[|B_1|^2]\mathbb{E}[Y_1^2]\right)^{1/2} + \frac{4}{\sqrt{n}}\mathbb{E}[Y_1^4]^{1/2}.\end{aligned} \tag{72}$$

Recall that A_1 has a multivariate t-distribution $t_\nu(0, \mathbb{1}_d)$, hence $A_1 \overset{d}{=} Z/\sqrt{U/\nu}$ where $Z \sim \mathcal{N}(0, \mathbb{1}_d)$ and $U \sim \chi^2(\nu)$ are independent. Thus, $\mathbb{E}[\|A_1\|^2] = \mathbb{E}[\|Z\|^2]\mathbb{E}[\nu/U] = \nu d/(\nu-2)$. Using that $\|X_1\|_\infty \leq M$ and $\lambda \leq \frac{C_{\text{lam}} d^p}{\sqrt{N}}$ we may thus deduce from (72) that

$$\begin{aligned}&\mathbb{E}[\mathcal{R}(H^{A,B}_{\widehat{W}_\lambda}) - \mathcal{R}_n(H^{A,B}_{\widehat{W}_\lambda}) + \mathcal{R}_n(H^{A,B}_{W}) - \mathcal{R}(H^{A,B}_{W})] \\ &\leq \frac{C_{\text{est}}^2 d^{2p+2}}{\sqrt{n}}\end{aligned} \tag{73}$$

with $C_{\text{est}}^2 = 8C_{\text{lam}}^2(\frac{\nu}{\nu-2}M^2 + \mathbb{E}[|B_1|^2]) + 2^{3+\frac{1}{2}}C_{\text{lam}}(\frac{\nu}{\nu-2}M^2\mathbb{E}[Y_1^2] + \mathbb{E}[|B_1|^2]\mathbb{E}[Y_1^2])^{1/2} + 4\mathbb{E}[Y_1^4]^{1/2}$ not depending on d, n or N. Combining (73) with (64) and (65) we obtain

$$\mathbb{E}[|H(\bar{X}) - H^{A,B}_{\widehat{W}_\lambda}(\bar{X})|^2] \leq \left(\frac{C_{\text{app}}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu+d)^{k+3}}{\sqrt{N}}\right)^2 + \frac{C_{\text{est}}^2 d^{2p+2}}{\sqrt{n}}. \tag{74}$$

■

4.4 Stochastic Gradient Descent

For the most common choices of $\mathcal{W}$ the solution to the optimization problem (54) can be obtained by solving the system of linear equations (56) or (61), respectively. There may nevertheless be situations in which one is interested in solving (54) using a stochastic gradient descent method (e.g. when comparing the performance of different learning methods in an experiment). Therefore, we will briefly discuss optimization of (60) by stochastic gradient descent here and combine our error bound in Theorem 19 with the stochastic gradient descent optimization error bound from Shamir and Zhang (2013).

To this end, let $\mathcal{V} = \{w \in \mathbb{R}^N \,|\, \|w\| \leq \lambda\}$ denote the set within which we look for an optimizer, let $\Pi_{\mathcal{V}} \colon \mathbb{R}^N \to \mathcal{V}$ be the orthogonal projection onto $\mathcal{V}$, for $i = 1, \ldots, n$ write $\mathbf{X}^i$ for the $\mathbb{R}^N$-valued random vector with components $\mathbf{X}^i_j = \varrho(A_j \cdot X_i + B_j)$, $j = 1, \ldots, N$, let $\mathcal{T} \in \{2, 3, \ldots\}$ denote the number of stochastic gradient descent iterations, let $\mathfrak{B} \in \{1, \ldots, n\}$ denote the batch size and let $J = \{J_{i,t}\}_{(i,t) \in \{1,\ldots,\mathfrak{B}\} \times \{1,\ldots,\mathcal{T}\}}$ denote i.i.d. random variables each having a uniform distribution on $\{1, \ldots, n\}$ and independent of $(A, B, D_n, \bar{X}, \bar{Y})$. Then, starting with $W_1 = 0$, we iteratively compute

$$W_{t+1} = \Pi_{\mathcal{V}} \left(W_t - \frac{2\eta_t}{\mathfrak{B}} \sum_{i=1}^{\mathfrak{B}} \mathbf{X}^{J_{i,t}} (W_t \cdot \mathbf{X}^{J_{i,t}} - Y_{J_{i,t}}) \right), \quad t = 1, \ldots, \mathcal{T} - 1, \tag{75}$$

where $\eta_t = \eta_0 t^{-1/2}$ for $t = 1, \ldots, \mathcal{T} - 1$. The parameter vector $W_{\mathcal{T}}$ is then used for the random neural network, i.e., $H^{A,B}_{W_{\mathcal{T}}}$ is the learned function approximating H. The next proposition provides a bound on the prediction error.

Proposition 21 *Let $C > \frac{1}{2^{3/2}\pi}$, $\eta_0 > 0$ and $\nu > 4$. Suppose $A_1 \sim t_\nu(0, \mathbb{1}_d)$ and B_1 has density π_b satisfying (26). Suppose $H \colon \mathbb{R}^d \to \mathbb{R}$ is of the form $H(x) = \mathbb{E}[\Phi(x + V)]$ with $\Phi \in L^1(\mathbb{R}^d)$ and V satisfying (25). Assume that $\|X_1\|_\infty \leq M$, $\mathbb{P}$-a.s. and $\mathbb{E}[|Y_1|^4] < \infty$. Let $\eta_t = \eta_0 t^{-1/2}$ for $t = 1, \ldots, \mathcal{T} - 1$ and $\lambda \in \frac{1}{\sqrt{N}}[C_{wgt}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu + d)^{2k + \frac{1}{2}}, C_{lam} d^p]$ with $k \in \mathbb{N}$, $C_{wgt} > 0$ as in Theorem 7 and $p \geq 0$, $C_{lam} > 0$ not depending on n, N, d or $\mathcal{T}$.*

Then there exist $C_{app}, C_{est}, C_{opt} > 0$ such that

$$\begin{aligned} \mathbb{E}[|H(\bar{X}) - H^{A,B}_{W_{\mathcal{T}}}(\bar{X})|^2]^{1/2} \leq \frac{C_{app}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu + d)^{k+3}}{\sqrt{N}} + \frac{C_{est} d^{p+1}}{n^{\frac{1}{4}}} \\ + \frac{C_{opt} d^{p+2} N (2 + \log(\mathcal{T}))^{\frac{1}{2}}}{\mathcal{T}^{\frac{1}{4}}}. \end{aligned} \tag{76}$$

The constant k only depends on π_b and the constants $C_{app}, C_{est}, C_{opt}$ depend on ν, π_b, C, M, $\mathbb{E}[Y_1^4], \eta_0, C_{lam}$, but they do not depend on d, n, N or $\mathcal{T}$.

Remark 22 *The first two terms in the error bound in (76) are as in the bound (62) in Theorem 19, whereas the last term in (76) is due to the stochastic gradient descent optimization. The rate of convergence to 0 of this last error term as a function of $\mathcal{T}$ could be further improved, e.g., by using a more refined optimization scheme (based on averaging) than (75), see for instance Shamir and Zhang (2013). However, for our purposes the bound in Proposition 21 suffices as this bound already proves that the overall error does not suffer from the curse of dimensionality.*

Proof Let $C_{\text{app}} > 0$ be as in Theorem 7, let W be the $\mathbb{R}^N$-valued, $\sigma(A,B)$-measurable random vector satisfying (27) (see Theorem 7) and let $C_{\text{est}} > 0$ be as in Theorem 19.

By independence and (51) we obtain (as in (63)-(64) in the proof of Theorem 19)

$$\begin{aligned}
&\mathbb{E}[|H(\bar{X}) - H^{A,B}_{W_{\mathcal{T}}}(\bar{X})|^2] \\
&\quad = \mathbb{E}[|H(\bar{X}) - H^{A,B}_{W}(\bar{X})|^2] + \mathbb{E}[\mathcal{R}(H^{A,B}_{W_{\mathcal{T}}}) - \mathcal{R}(H^{A,B}_{W})] \\
&\quad \leq \mathbb{E}[|H(\bar{X}) - H^{A,B}_{W}(\bar{X})|^2] \\
&\qquad + \mathbb{E}[\mathcal{R}(H^{A,B}_{W_{\mathcal{T}}}) - \mathcal{R}_n(H^{A,B}_{W_{\mathcal{T}}}) + \mathcal{R}_n(H^{A,B}_{W_{\mathcal{T}}}) - \mathcal{R}_n(H^{A,B}_{\widehat{W}_\lambda}) + \mathcal{R}_n(H^{A,B}_{W}) - \mathcal{R}(H^{A,B}_{W})],
\end{aligned} \tag{77}$$

where we used (60) and $W \in \mathcal{W}_\lambda$ (as established in the proof of Theorem 19) in the last step. The first expectation in the right hand side of (77) has been bounded in (65) in the proof of Theorem 19. For the second expectation we may proceed analogously as in (66): we use the same notation as in (66) and, in addition, write $W^{a,b}_{\mathcal{T}}$ for the output of the stochastic gradient descent algorithm with (A,B) fixed to (a,b). Then independence yields

$$\begin{aligned}
&\mathbb{E}[\mathcal{R}(H^{A,B}_{W_{\mathcal{T}}}) - \mathcal{R}_n(H^{A,B}_{W_{\mathcal{T}}}) + \mathcal{R}_n(H^{A,B}_{W}) - \mathcal{R}(H^{A,B}_{W})] \\
&\quad = \mathbb{E}[\mathbb{E}[\mathcal{R}(H^{a,b}_{W^{a,b}_{\mathcal{T}}}) - \mathcal{R}_n(H^{a,b}_{W^{a,b}_{\mathcal{T}}}) + \mathcal{R}_n(H^{a,b}_{W^{a,b}}) - \mathcal{R}(H^{a,b}_{W^{a,b}})]\Big|_{(a,b)=(A,B)}] \\
&\quad \leq 2\mathbb{E}\left[\mathbb{E}\left[\sup_{w\in\mathcal{W}^0_\lambda}\left|\mathcal{R}(H^{a,b}_w) - \mathcal{R}_n(H^{a,b}_w)\right|\right]\Big|_{(a,b)=(A,B)}\right].
\end{aligned} \tag{78}$$

Now we can compare (77) and (78) to (64) and (66) in the proof of Theorem 19. We see that the decomposition (77) yields the same error terms as in Theorem 19 plus the additional term $\mathbb{E}[\mathcal{R}_n(H^{A,B}_{W_{\mathcal{T}}}) - \mathcal{R}_n(H^{A,B}_{\widehat{W}_\lambda})]$.

Therefore, Theorem 19 shows that

$$\begin{aligned}
\mathbb{E}[|H(\bar{X}) - H^{A,B}_{W_{\mathcal{T}}}(\bar{X})|^2]^{1/2} \leq \frac{C_{\text{app}}\|\Phi\|_{L^1(\mathbb{R}^d)}(\nu+d)^{k+3}}{\sqrt{N}} + \frac{C_{\text{est}}d^{p+1}}{n^{\frac{1}{4}}} \\
+ \mathbb{E}[\mathcal{R}_n(H^{A,B}_{W_{\mathcal{T}}}) - \mathcal{R}_n(H^{A,B}_{\widehat{W}_\lambda})]^{1/2}.
\end{aligned} \tag{79}$$

We now analyze the last term. Write $W^{a,b,d_n}_{\mathcal{T}}$ for the output of the stochastic gradient descent algorithm and $\widehat{W}^{a,b,d_n}_{\lambda}$ for the solution to (60) when $(A,B,D_n) = (a,b,d_n)$. From the updating scheme it is clear that there exists a measurable function F such that $W_{\mathcal{T}} = F(A,B,D_n,J) = W^{A,B,D_n}_{\mathcal{T}}$. Furthermore (as argued in the proof of Theorem 19), $\widehat{W}^{a,b,d_n}_{\lambda} = G(a,b,d_n)$ for a measurable function G and $\widehat{W}^{A,B,D_n}_{\lambda} = \widehat{W}_\lambda$. Thus, we may use independence to write

$$\mathbb{E}[\mathcal{R}_n(H^{A,B}_{W_{\mathcal{T}}}) - \mathcal{R}_n(H^{A,B}_{\widehat{W}_\lambda})] = \mathbb{E}[\mathbb{E}[\mathcal{R}^{d_n}_n(H^{a,b}_{W^{a,b,d_n}_{\mathcal{T}}}) - \mathcal{R}^{d_n}_n(H^{a,b}_{\widehat{W}^{a,b,d_n}_{\lambda}})]\Big|_{(a,b,d_n)=(A,B,D_n)}], \tag{80}$$

where $\mathcal{R}^{d_n}_n(f) = \frac{1}{n}\sum_{i=1}^n (f(x_i) - y_i)^2$ for $d_n = ((x_1,y_1),\dots,(x_n,y_n))$. Consider $(a,b,d_n) \in (\mathbb{R}^d)^N \times \mathbb{R}^N \times ([-M,M]^d \times \mathbb{R})^n$ as fixed now and write $\mathbf{x}^i$ for the vector with $\mathbf{x}^i_j = \varrho(a_j \cdot x_i + b_j)$,

$j = 1, \dots, N$. Let $F\colon \mathcal{V} \to \mathbb{R}$, $F(w) = \frac{1}{n}\sum_{i=1}^{n}(w \cdot \mathbf{x}^i - y_i)^2$. Then $H_w^{a,b}(x_i) = w \cdot \mathbf{x}^i$, $\mathcal{R}_n^{d_n}(H_w^{a,b}) = F(w)$ and hence $\hat{w} := \widehat{W}_\lambda^{a,b,d_n}$ is a (global) minimizer of F in $\mathcal{V}$. Write $w_t := W_t^{a,b,d_n}$ and recall

$$w_{t+1} = \Pi_{\mathcal{V}}\left(w_t - \eta_t \hat{g}_t\right), \quad t = 1, \dots, \mathcal{T} - 1 \tag{81}$$

with $\hat{g}_t = \frac{2}{\mathfrak{B}}\sum_{i=1}^{\mathfrak{B}} \mathbf{x}^{J_{i,t}}(w_t \cdot \mathbf{x}^{J_{i,t}} - y_{J_{i,t}})$. Independence implies $\mathbb{E}[\hat{g}_t | w_t] = \frac{2}{\mathfrak{B}}\sum_{i=1}^{\mathfrak{B}} \mathbb{E}[\mathbf{x}^{J_{i,t}}(w \cdot \mathbf{x}^{J_{i,t}} - y_{J_{i,t}})]|_{w=w_t} = \frac{2}{n}\sum_{j=1}^{n} \mathbf{x}^j(w_t \cdot \mathbf{x}^j - y_j) = \nabla F(w_t)$. Furthermore, F is convex and the Minkowski integral inequality and independence yield

$$\begin{aligned}
\mathbb{E}[\|\hat{g}_t\|^2] &\leq 4\mathbb{E}\left[\left(\frac{1}{\mathfrak{B}}\sum_{i=1}^{\mathfrak{B}} \|\mathbf{x}^{J_{i,t}}\|(|w_t \cdot \mathbf{x}^{J_{i,t}}| + |y_{J_{i,t}}|)\right)^2\right] \\
&\leq 4\left(\frac{1}{\mathfrak{B}}\sum_{i=1}^{\mathfrak{B}} \left(\mathbb{E}\left[\|\mathbf{x}^{J_{i,t}}\|^2(|w_t \cdot \mathbf{x}^{J_{i,t}}| + |y_{J_{i,t}}|)^2\right]\right)^{1/2}\right)^2 \\
&\leq \frac{8}{n}\sum_{j=1}^{n} \|\mathbf{x}^j\|^2(\mathbb{E}[\|w_t\|^2]\|\mathbf{x}^j\|^2 + |y_j|^2) \\
&\leq \frac{16}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{N} \|a_j\|^2\|x_i\|^2 + |b_j|^2\right)(\lambda^2\|\mathbf{x}^i\|^2 + |y_i|^2) \\
&\leq 32\left(1 + M^2 d\|a\|_F^2 + \|b\|^2\right)^2(\lambda^2 + \frac{1}{n}\sum_{i=1}^{n}|y_i|^2),
\end{aligned} \tag{82}$$

where in the last two inequalities we used the estimate $\left\|\mathbf{x}^i\right\|^2 = \sum_{j=1}^{N}[\varrho(a_j \cdot x_i + b_j)]^2 \leq 2\sum_{j=1}^{N}\|a_j\|^2\|x_i\|^2 + |b_j|^2$. Shamir and Zhang (2013, Theorem 2) hence implies that

$$\mathbb{E}[F(w_{\mathcal{T}}) - F(\hat{w})] \leq \left(\frac{4\lambda^2}{\eta_0} + \eta_0 32\left(1 + dM^2\|a\|_F^2 + \|b\|^2\right)^2(\lambda^2 + \frac{1}{n}\sum_{i=1}^{n}|y_i|^2)\right)\frac{2 + \log(\mathcal{T})}{\sqrt{\mathcal{T}}}.$$

Inserting this in (80) and using independence yields

$$\begin{aligned}
&\mathbb{E}[\mathcal{R}_n(H_{W_{\mathcal{T}}}^{A,B}) - \mathcal{R}_n(H_{\widehat{W}_\lambda}^{A,B})] \\
&\quad\leq \mathbb{E}\left[\frac{4\lambda^2}{\eta_0} + 32\eta_0\left(1 + dM^2\|A\|_F^2 + \|B\|^2\right)^2(\lambda^2 + \frac{1}{n}\sum_{i=1}^{n}|Y_i|^2)\right]\frac{2 + \log(\mathcal{T})}{\sqrt{\mathcal{T}}} \\
&\quad\leq \left(\frac{4\lambda^2}{\eta_0} + 96\eta_0(1 + d^2M^4\mathbb{E}[\|A\|_F^4] + \mathbb{E}[\|B\|^4])(\lambda^2 + \mathbb{E}[|Y_1|^2])\right)\frac{2 + \log(\mathcal{T})}{\sqrt{\mathcal{T}}}.
\end{aligned} \tag{83}$$

Employing Minkowski's integral inequality we estimate

$$\begin{aligned}
d^2M^4\mathbb{E}[\|A\|_F^4] + \mathbb{E}[\|B\|^4] &\leq d^2M^4\left(\sum_{j=1}^{N}\mathbb{E}[\|A_j\|^4]^{1/2}\right)^2 + \left(\sum_{j=1}^{N}\mathbb{E}[|B_j|^4]^{1/2}\right)^2 \\
&= N^2(d^2M^4\mathbb{E}[\|A_1\|^4] + \mathbb{E}[|B_1|^4]).
\end{aligned} \tag{84}$$

Recall that $A_1 \overset{d}{=} Z/\sqrt{U/\nu}$, where $Z \sim \mathcal{N}(0, \mathbb{1}_d)$ and $U \sim \chi^2(\nu)$ are independent. Therefore $\mathbb{E}[\|A_1\|^4] = \mathbb{E}[\|Z\|^4]\mathbb{E}[\nu^2/U^2]$ and one obtains analogously to (84) the estimate $\mathbb{E}[\|Z\|^4] \leq d^2\mathbb{E}[Z_1^4]$. Inserting this into (84) and (83) and estimating $\lambda \leq \frac{C_{\text{lam}} d^p}{\sqrt{N}}$ yields

$$\begin{aligned}
&\mathbb{E}[\mathcal{R}_n(H^{A,B}_{W_\mathcal{T}}) - \mathcal{R}_n(H^{A,B}_{\widehat{W}_\lambda})] \\
&\quad \leq \frac{C_{\text{opt}}^2}{4}\left((1+N^2)\frac{d^{2p+4}}{N} + (1+N^2)d^4\right)\frac{2+\log(\mathcal{T})}{\sqrt{\mathcal{T}}} \\
&\quad \leq C_{\text{opt}}^2 d^{2p+4} N^2 \frac{2+\log(\mathcal{T})}{\sqrt{\mathcal{T}}}
\end{aligned} \tag{85}$$

with $C_{\text{opt}}^2 = 4\max(\frac{4}{\eta_0}, 96\eta_0)\max(2, 3M^4\nu^2/[(\nu-2)(\nu-4)] + \mathbb{E}[|B_1|^4])\max(C_{\text{lam}}^2, \mathbb{E}[|Y_1|^2])$ and where we used $\mathbb{E}[Z_1^4] = 3$, $\mathbb{E}[U^{-2}] = 1/[(\nu-2)(\nu-4)]$. Combining this with (79) yields (76), as claimed. ∎

4.5 Application to Basket Option Pricing

As a first application of the results derived in Sections 4.2–4.4 we consider the problem of learning prices of basket put options in certain "non-degenerate" models.

Suppose that y_i is the market price of a put option with strike $K_i > 0$ written on a basket of m assets. Assume that, up to some additive noise, these market prices are "generated" from an unknown, non-degenerate stochastic model. This means that we assume

$$y_i = \mathbb{E}\left[\max\left(K_i - \sum_{i=1}^m w_i S_{T,i}, 0\right)\right] + \varepsilon_i, \quad i = 1, \dots, n,$$

where $\varepsilon_1, \dots, \varepsilon_n$ are i.i.d. random variables, $S_T = (S_{T,1}, \dots, S_{T,m})$ is a $[0,\infty)^m$-valued random vector and $w_1, \dots, w_m \in [0,\infty)$ are non-negative weights. Assume that $\mathbb{E}[\varepsilon_1] = 0$, $\mathbb{E}[\varepsilon_1^4] < \infty$ and $\{\varepsilon_i\}_{i=1,\dots,n}$ are independent of (A, B, S_T). We think of S_T as the value at time T of a price process S (for which $\mathbb{P}$ is a martingale measure).

The goal is to learn the pricing function $H(K) := \mathbb{E}\left[\max\left(K - \sum_{i=1}^m w_i S_{T,i}, 0\right)\right]$ from the observed market prices $y_1, \dots, y_n$.

This fits into the framework introduced above (see Section 4.1) if we let $M = \max_{i=1,\dots,n} K_i$ and consider $K_1, \dots, K_n$ as the observed realizations of the n i.i.d. random variables $X_1, \dots, X_n$ so that also $y_i = H(K_i) + \varepsilon_i$ is the realization of $Y_i = H(X_i) + \varepsilon_i$. We assume that X_1 is distributed uniformly on $[0, M]$ and $\{X_i\}_{i=1,\dots,n}$ are independent of $\{\varepsilon_i\}_{i=1,\dots,n}, (A, B)$. Then the option pricing function H is indeed the regression function (51) and we obtain the following corollary. Recall that $\bar{X}$ has the same distribution as X_1 and is independent of (A, B, D_n).

Corollary 23 *Let $\nu > 4$, $C > \frac{1}{2^{3/2}\pi}$, $\eta_0 > 0$ and $\bar{c} > 0$ be constants which do not depend on n, N or $\mathcal{T}$. Suppose $A_1 \sim t_\nu(0,1)$ and B_1 has density π_b satisfying (26). Assume that the $[0,\infty)^m$-valued random vector S_T satisfies $|\mathbb{E}[e^{-i\xi w\cdot S_T}]| \leq \exp(-C|\xi|^2)$ for all $\xi \in \mathbb{R}$. Then*

there exists $C_0 > 0$ such that the prediction error bound

$$\mathbb{E}[|H(\bar{X}) - T_M(H^{A,B}_{\widehat{W}}(\bar{X}))|^2]^{1/2} \leq C_0 \left(\frac{(\log(n)+1)^{1/2}\sqrt{N}}{\sqrt{n}} + \frac{1}{\sqrt{N}} \right) \tag{86}$$

holds and there exist $C_1, C_2, \underline{c} > 0$ such that for any $\lambda \in \frac{1}{\sqrt{N}}[\underline{c}, \bar{c}]$ the prediction error bounds

$$\mathbb{E}[|H(\bar{X}) - H^{A,B}_{\widehat{W}_\lambda}(\bar{X})|^2]^{1/2} \leq C_1 \left(\frac{1}{\sqrt{N}} + \frac{1}{n^{\frac{1}{4}}} \right), \tag{87}$$

$$\mathbb{E}[|H(\bar{X}) - H^{A,B}_{W_\mathcal{T}}(\bar{X})|^2]^{1/2} \leq C_2 \left(\frac{1}{\sqrt{N}} + \frac{1}{n^{\frac{1}{4}}} + \frac{N(2+\log(\mathcal{T}))^{\frac{1}{2}}}{\mathcal{T}^{\frac{1}{4}}} \right) \tag{88}$$

hold. The constants $C_0, C_1, C_2, \underline{c}$ do not depend on n, N or $\mathcal{T}$.

Remark 24 *The proof of Corollary 23 shows that $\underline{c}$ does not depend on $\bar{c}$. Hence, by choosing $\bar{c} > \underline{c}$ it can always be guaranteed that $[\underline{c}, \bar{c}]$ is not empty.*

Remark 25 *The hypothesis $|\mathbb{E}[e^{-i\xi w \cdot S_T}]| \leq \exp(-C|\xi|^2)$ is inherited from Theorem 7. In Theorem 7 this hypothesis guarantees that the constants do not grow exponentially in the dimension d. In the situation here $d = 1$ and so this hypothesis could be relaxed considerably: it could be replaced by the assumption $|\mathbb{E}[e^{-i\xi w \cdot S_T}]| \leq \exp(-C|\xi|^\alpha)$ for some $C > 0$, $\alpha > 0$ or even by the assumption that $|\mathbb{E}[e^{-i\xi w \cdot S_T}]| \leq C(1+|\xi|)^{-\beta}$ for some $C > 0$ and sufficiently large $\beta > 0$ (depending on ν and π_b).*

Proof Firstly, by assumption we have $|X_1| \leq M$, $\mathbb{P}$-a.s. and $H\colon \mathbb{R} \to \mathbb{R}$ satisfies for $K \in [0, M]$ that

$$H(K) = \mathbb{E}\left[\max\left(K - w \cdot S_T, 0\right)\right] = \mathbb{E}[\Phi(K+V)]$$

with $V = -w \cdot S_T$ and $\Phi(y) = y\mathbb{1}_{[0,M]}(y)$. Hence, $H(\bar{X}) = \tilde{H}(\bar{X})$ $\mathbb{P}$-a.s. with $\tilde{H}(x) = \mathbb{E}[\Phi(x+V)]$ for $x \in \mathbb{R}$. Furthermore, $\Phi \in L^1(\mathbb{R})$, V satisfies (25), $\sigma^2 = \sup_{x\in\mathbb{R}} \mathbb{E}[(Y_1 - H(X_1))^2 | X_1 = x] = \mathbb{E}[\varepsilon_1^2] < \infty$ and $|\tilde{H}(x)| \leq M$ for all $x \in \mathbb{R}$. Thus, the hypotheses of Theorem 17 with $L = M$ are satisfied and so, using $H(\bar{X}) = \tilde{H}(\bar{X})$ $\mathbb{P}$-a.s., we obtain that there exist $k \in \mathbb{N}$ and $\tilde{C}_{\text{app}} > 0$ such that the prediction error bound (57) holds. Hence (86) follows with $C_0 = \max(\tilde{C}_{\text{app}} \max(\sigma, M), \tilde{C}_{\text{app}} \|\Phi\|_{L^1(\mathbb{R})}(\nu+1)^{k+3})$.

Next we prove (87). To this end, notice $\mathbb{E}[|Y_1|^4] \leq 8(\mathbb{E}[|H(X_1)|^4] + \mathbb{E}[|\varepsilon_1|^4]) \leq 8(M^4 + \mathbb{E}[|\varepsilon_1|^4]) < \infty$ and let $C_{\text{app}}, C_{\text{wgt}} > 0$ be as in Theorem 7. Then Theorem 19 proves that for any $\lambda > 0$ satisfying $\frac{C_{\text{wgt}}\|\Phi\|_{L^1(\mathbb{R})}(\nu+1)^{2k+\frac{1}{2}}}{\sqrt{N}} \leq \lambda \leq \frac{\bar{c}}{\sqrt{N}}$ there exists a constant $C_{\text{est}} > 0$ such that the prediction error bound (62) holds. The proof actually shows that the same constant can be chosen for all λ in the specified range. Thus, (87) follows with $C_1 = \max(C_{\text{app}}\|\Phi\|_{L^1(\mathbb{R})}(\nu+1)^{k+3}, C_{\text{est}})$ and $\underline{c} = C_{\text{wgt}}\|\Phi\|_{L^1(\mathbb{R})}(\nu+1)^{2k+\frac{1}{2}}$.

Furthermore, Proposition 21 proves that there exists $C_{\text{opt}} > 0$ such that (76) holds. Setting $C_2 = \max(C_1, C_{\text{opt}})$ we obtain (88).

In these results we proved that the constants $C_{\text{app}}, C_{\text{est}}, C_{\text{opt}}$ depend on $\nu, \pi_{\text{b}}, C, M$, $\mathbb{E}[Y_1^4]$, η_0, $\bar{c}$, but they do not depend on n, N or $\mathcal{T}$, hence it follows that $C_0, C_1, C_2, \underline{c}$ do not depend on n, N or $\mathcal{T}$.

■

5. Learning Black-Scholes Type PDEs

In this section we apply the results from Section 4 to prove that random neural networks are capable of learning certain Black-Scholes type partial (integro-)differential equations (also referred to as (non-local) PDEs) without the curse of dimensionality. More specifically, we consider the problem of learning solutions to Kolmogorov PDEs associated to exponential Lévy-processes, which includes the Black-Scholes PDE as a special case. The learning methods used to tackle this problem are random neural networks trained by (constrained) regression or stochastic gradient descent. By combining the results from Theorems 13, 17, 19 and from Proposition 21 we obtain bounds on the prediction error. The dependence on the dimension d in these bounds is explicit and at most polynomial, whereas the bounds decay at polynomial rate in the number of samples n and the network size N (and the number of stochastic gradient descent iterations $\mathcal{T}$). Hence, the number of samples, hidden nodes of the network and gradient steps required to achieve a prescribed prediction accuracy $\varepsilon > 0$ grows at most polynomially in d and ε^{-1}. This means that random neural networks are capable of learning solutions to such Kolmogorov PDEs without the curse of dimensionality.

For the reader's convenience we introduce in Section 5.1 in detail again all the objects relevant to the discussion. Section 5.2 then contains the prediction error bounds for Black-Scholes type PDEs. We conclude in Section 5.4 with numerical experiments.

5.1 Formulation of the Learning Problem for PDEs

We again put ourselves in the situation studied in Section 3.2 and consider for each $d \in \mathbb{N}$ the partial (integro-)differential equation

$$\begin{aligned} \partial_t u_d(t,s) &= \tfrac{1}{2}\sum_{k,l=1}^d s_k s_l \Sigma^d_{k,l} \partial_{s_k}\partial_{s_l} u_d(t,s) + \sum_{i=1}^d s_i \tilde{\gamma}^d_i \partial_{s_i} u_d(t,s) \\ &\quad + \int_{\mathbb{R}^d} \Big[u_d(t, se^y) - u_d(t,s) - \sum_{i=1}^d (e^{y_i}-1) s_i \partial_{s_i} u_d(t,s) \Big] \nu^d_{\mathrm{L}}(dy), \qquad (89) \\ u_d(0,s) &= \varphi_d(s) \end{aligned}$$

for $s \in (0,\infty)^d, t > 0$, where $\varphi_d \colon (0,\infty)^d \to \mathbb{R}$ is a "payoff" function and $(\Sigma^d, \gamma^d, \nu^d_{\mathrm{L}})$ is the characteristic triplet of a Lévy process L^d, we write $\tilde{\gamma}^d_i = \gamma^d_i + \frac{1}{2}\Sigma^d_{i,i} + \int_{\mathbb{R}^d}(e^{y_i} - 1 - y_i \mathbb{1}_{\{\|y\|\leq 1\}})\nu^d_{\mathrm{L}}(dy)$, $i = 1,\dots,d$, for the shifted drift vector and we assume $\nu^d_{\mathrm{L}}(\{y \in \mathbb{R}^d \,|\, \|y\| > R\}) = 0$ for some $R > 1$. Furthermore, we recall the notation $s\exp(x) = (s_1 \exp(x_1), \dots, s_d \exp(x_d))$ for $s, x \in \mathbb{R}^d$.

The (non-local) PDE (89) is the Kolmogorov PDE for the exponential Lévy model associated to L^d, see Section 3.2 for further interpretation and a discussion on the relation to option pricing and the assumption on ν^d_{L}. If $\nu^d_{\mathrm{L}} = 0$, then (89) is the Black-Scholes PDE.

Let $T > 0$ and suppose we are given i.i.d. $\mathbb{R}^d \times \mathbb{R}$-valued random variables $(X^d_1, Y^d_1), (X^d_2, Y^d_2), \dots$ with the property that

$$u_d(T, \exp(x)) = \mathbb{E}[Y^d_1 | X^d_1 = x], \qquad (90)$$

for $(\mathbb{P} \circ (X^d_1)^{-1})$-a.e. $x \in \mathbb{R}^d$, that is, $u_d(T, \exp(\cdot))$ is the regression function. We are interested in learning $u_d(T,\cdot)$ on the set $\mathcal{D}^d = \{\exp(x) \,|\, x \in [-M, M]^d\} \subset (0,\infty)^d$. This encompasses two particularly relevant situations.

Example 1 *Suppose that the solution $u_d(T,\cdot)$ of the PDE can be observed at n points $\exp(X^d_1), \dots, \exp(X^d_n)$. The observations are not perfect, but perturbed by some additive*

noise. The goal is to learn the solution of the PDE on the entire set $\mathcal{D}^d$ from these noisy observations. This situation is captured in our setting with $Y_i^d = u_d(T, \exp(X_i^d)) + \varepsilon_i^d$ for $i = 1, \dots, n$, where $\varepsilon_1^d, \dots, \varepsilon_n^d$ are i.i.d. random variables independent of $X_1^d, \dots, X_n^d$.

Example 2 *A different situation of interest arises when neural networks are employed as a solution method for the PDE* (89) *in the way proposed in Berner et al. (2020) for a related setting. Let $X_1^d, \dots, X_n^d$ be i.i.d. random variables uniformly distributed on $[-M, M]^d$ and independent of L^d and let $Y_i^d = \varphi_d(\exp(X_i^d + L_T^d))$ for $i = 1, \dots, n$. Then one may show using the Feynman-Kac formula (see Proposition 16) that*

$$u_d(T, \exp(x)) = \mathbb{E}[\varphi_d(\exp(x + L_T^d))] = \mathbb{E}[\varphi_d(\exp(X_1^d + L_T^d))|X_1^d = x] = \mathbb{E}[Y_1^d|X_1^d = x]$$

for $(\mathbb{P} \circ (X_1^d)^{-1})$-a.e. $x \in \mathbb{R}^d$ and hence $u_d(T, \exp(\cdot))$ is indeed the regression function (90)*. Thus, in this situation we have formulated the problem of solving the PDE* (89) *on $\mathcal{D}^d$ as a statistical learning problem with data points $(X_i, \varphi_d(\exp(X_i^d + L_T^d)))$, $i = 1, \dots, n$.*

In order to learn the unknown function $u_d(T, \cdot)$ from the data $D_n^d = ((X_1^d, Y_1^d), \dots, (X_n^d, Y_n^d))$ we employ a random neural network. Recall from Section 2 that a random neural network is a single-hidden-layer feedforward neural network in which the hidden weights are randomly generated and then considered fixed and only the output-layer weight vector can be trained. The weights of the random neural networks are generated as follows: let $\nu > 4$, for each $d \in \mathbb{N}$ let $A_1^d, A_2^d, \dots$ be i.i.d. $\mathbb{R}^d$-valued random vectors and let $B_1, B_2, \dots$ be i.i.d. random variables. Assume that A_1^d is $t_\nu(0, \mathbb{1}_d)$-distributed and B_1 has a strictly positive Lebesgue-density π_{b} of at most polynomial decay (see (26)). For each $d, n \in \mathbb{N}$ we assume that $\{A_i^d\}_{i\in\mathbb{N}}$, $\{B_i\}_{i\in\mathbb{N}}$ and D_n^d are independent. For $d, N \in \mathbb{N}$ we write $A^{d,N} = (A_1^d, \dots, A_N^d)$ and $B^N = (B_1, \dots, B_N)$. If N hidden nodes are used, the random neural network employed for learning is then given by

$$H_W^{A^{d,N},B^N}(x) = \sum_{i=1}^{N} W_i \varrho(A_i^d \cdot x + B_i), \quad x \in \mathbb{R}^d, \tag{91}$$

where W is an $\mathbb{R}^N$-valued, $\sigma(A^{d,N}, B^N, D_n^d)$-measurable random vector which needs to be chosen. The (squared) learning error (or prediction error) is given by

$$\mathbb{E}[|u_d(T, \exp(\bar{X}^d)) - H_W^{A^{d,N},B^N}(\bar{X}^d)|^2], \tag{92}$$

where $(\bar{X}^d, \bar{Y}^d)$ has the same distribution as (X_1^d, Y_1^d) and is independent of $\{(A_i^d, B_i)\}_{i\in\mathbb{N}}$ and D_n^d.

Learning $u_d(T, \cdot)$ by $H_W^{A^{d,N},B^N}$ then amounts to selecting an $\mathbb{R}^N$-valued (random) vector W that minimizes the prediction error. W may be chosen depending on the random weights $A^{d,N}, B^N$ and the data $D_n^d = ((X_1^d, Y_1^d), \dots, (X_n^d, Y_n^d))$. We consider three choices:

- W is chosen as $\widehat{W}^{d,N,n}$, where

$$\widehat{W}^{d,N,n} = \arg \min_{W \in \mathcal{W}^{d,N,n}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (H_W^{A^{d,N},B^N}(X_i^d) - Y_i^d)^2 \right\} \tag{93}$$

  for $\mathcal{W}^{d,N,n} = \{W \colon \Omega \to \mathbb{R}^N \,|\, W \text{ is } \sigma(A^{d,N}, B^N, D_n^d)\text{-measurable}\}$. Note that $\widehat{W}^{d,N,n}$ can be calculated explicitly by solving a system of linear equations (see Section 4.2).

- W is chosen as $\widehat{W}_\lambda^{d,N,n}$, where

$$\widehat{W}_\lambda^{d,N,n} = \arg \min_{W \in \mathcal{W}_\lambda^{d,N,n}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (H_W^{A^{d,N},B^N}(X_i^d) - Y_i^d)^2 \right\} \tag{94}$$

  for $\mathcal{W}_\lambda^{d,N,n} = \{W \in \mathcal{W}^{d,N,n} \,|\, \|W\| \leq \lambda \ \mathbb{P}\text{-a.s.}\}$. Recall that $\widehat{W}_\lambda^{d,N,n}$ can be calculated explicitly by solving a system of linear equations (see Section 4.3).

- W is chosen as $W_{\mathcal{T}}^{d,N,n}$, where $W_{\mathcal{T}}^{d,N,n}$ is computed using the stochastic gradient descent algorithm as introduced in Section 4.4.

Remark 26 *As pointed out above, training of random neural networks can be performed by solving a system of linear equations (see* (56) *in Section 4.2 and* (61) *in Section 4.3). There may nevertheless be situations in which one is interested in training a random neural network using a stochastic gradient descent method (e.g. a performance comparison in an experiment). This is the reason why we also analyze optimization by stochastic gradient descent here.*

5.2 Learning Error Bounds

With these preparations (see Section 5.1) we now use the results from Sections 3 and 4 to provide sufficient conditions which guarantee that $u_d(T,\cdot)$ can be learnt using random neural networks without the curse of dimensionality.

Corollary 27 *Let $p \geq 0$, $c, L, M, \eta_0 > 0$, $C > \frac{1}{2^{3/2}T\pi}$. Assume that for each $d \in \mathbb{N}$ the payoff function satisfies $\varphi_d \circ \exp \in L^1(\mathbb{R}^d)$ and $\|\varphi_d \circ \exp\|_{L^1(\mathbb{R}^d)} \leq cd^p$, the characteristic triplet $(\Sigma^d, \gamma^d, \nu_{\mathrm{L}}^d)$ of the Lévy process L^d satisfies for all $\xi \in \mathbb{R}^d$*

$$\frac{1}{2}\xi \cdot \Sigma^d \xi \geq C\|\xi\|^2, \tag{95}$$

assume that $\|X_1^d\|_\infty \leq M$, $\mathbb{P}$-a.s. and suppose $u_d \in C^{1,2}((0,T)\times(0,\infty)^d) \cap C([0,T]\times(0,\infty)^d)$ is an at most polynomially growing solution to the PDE (89).

(i) Assume for all $d \in \mathbb{N}$ that $\sigma_d^2 = \sup_{x\in\mathbb{R}^d} \mathbb{E}[(Y_1^d - u_d(T, \exp(X_1^d)))^2 | X_1^d = x] \leq cd^p$ and $|u_d(T,s)| \leq L$ for all $s \in (0,\infty)^d$. Then there exist constants $C_0, \mathfrak{p} > 0$ such that for any $d, N, n \in \mathbb{N}$ the prediction error of random neural network regression satisfies

$$\mathbb{E}[|u_d(T, \exp(\bar{X}^d)) - T_L(H_{\widehat{W}^{d,N,n}}^{A^{d,N},B^N}(\bar{X}^d))|^2]^{1/2} \leq C_0 d^{\mathfrak{p}} \left(\frac{(\log(n)+1)^{1/2}\sqrt{N}}{\sqrt{n}} + \frac{1}{\sqrt{N}} \right). \tag{96}$$

(ii) Assume for all $d \in \mathbb{N}$ that $\mathbb{E}[|Y_1^d|^4] \leq cd^p$. Then there exist $\underline{p}, \underline{c} > 0$ such that for any $\overline{p} > \underline{p}$, $\overline{c} > \underline{c}$ there exist $C_0, \mathfrak{p} > 0$ such that for any $d, N, n \in \mathbb{N}$ the random neural network trained by constrained regression with parameter $\lambda \in \frac{1}{\sqrt{N}}[\underline{c}\underline{d}^{\underline{p}}, \overline{c}\overline{d}^{\overline{p}}]$ satisfies

$$\mathbb{E}[|u_d(T, \exp(\bar{X}^d)) - H_{\widehat{W}_\lambda^{d,N,n}}^{A^{d,N},B^N}(\bar{X}^d)|^2]^{1/2} \leq C_0 d^{\mathfrak{p}} \left(\frac{1}{\sqrt{N}} + \frac{1}{n^{\frac{1}{4}}} \right). \tag{97}$$

(iii) Consider the same situation as in (ii). Then, in addition, there exist constants $C_1, \mathfrak{q} > 0$ such that for any $d, N, n, \mathcal{T} \in \mathbb{N}$ the random neural network trained by stochastic gradient descent for $\mathcal{T}$ steps with learning rate $\eta_t = \eta_0 t^{-1/2}$ for $t = 1, \dots, \mathcal{T}-1$ and with λ as in (ii) satisfies

$$\mathbb{E}[|u_d(T, \exp(\bar{X}^d)) - H_{W_{\mathcal{T}}^{d,N,n}}^{A^{d,N},B^N}(\bar{X}^d)|^2]^{1/2} \leq C_1 d^{\mathfrak{q}} \left(\frac{1}{\sqrt{N}} + \frac{1}{n^{\frac{1}{4}}} + \frac{N(2+\log(\mathcal{T}))^{\frac{1}{2}}}{\mathcal{T}^{\frac{1}{4}}} \right). \tag{98}$$

Remark 28 *Each of these statements can be translated directly into a statement on the number of samples and hidden nodes required to guarantee a prescribed learning error of precision at most $\varepsilon > 0$. For instance, in the case of regression (corresponding to the bound (96)) we see that there exist constants $\tilde{C}_0, \tilde{\mathfrak{p}} > 0$ such that for all $d \in \mathbb{N}$, $\varepsilon > 0$ at most $N \leq \tilde{C}_0 d^{\tilde{\mathfrak{p}}} \varepsilon^{-2}$ weights and $n \leq \tilde{C}_0 d^{\tilde{\mathfrak{p}}} \varepsilon^{-8}$ samples suffice to guarantee*

$$\mathbb{E}[|u_d(T, \exp(\bar{X}^d)) - T_L(H_{\widehat{W}^{d,N,n}}^{A^{d,N},B^N}(\bar{X}^d))|^2]^{1/2} \leq \varepsilon. \tag{99}$$

This follows from (96) by choosing $N = 4C_0^2 d^{2\mathfrak{p}} \varepsilon^{-2}$, $n = 16c^2 C_0^4 d^{4\mathfrak{p}} \varepsilon^{-4} N^2$ and $\tilde{C}_0 = \max(4C_0^2, 256c^2C_0^8), \tilde{\mathfrak{p}} = 8\mathfrak{p}$ where c is a constant such that $\log(m) + 1 \leq c\sqrt{m}$ for all $m \in \mathbb{N}$.

Proof For fixed $d \in \mathbb{N}$ let $\Phi(x) = \varphi_d(\exp(x))$ and $H(x) = u_d(T, \exp(x))$ for $x \in \mathbb{R}^d$. Then Proposition 16 shows that $H(x) = \mathbb{E}[\Phi(x + L_T^d)]$ and, as argued in the proof of Theorem 13, the characteristic function of L_T^d satisfies the bound (46).

Proof of (i): Theorem 17 hence implies that there exist $k \in \mathbb{N}$ and $\tilde{C}_{\text{app}} > 0$ such that

$$\begin{aligned} &\mathbb{E}[|u_d(T, \exp(\bar{X}^d)) - T_L(H_{\widehat{W}^{d,N,n}}^{A^{d,N},B^N}(\bar{X}^d))|^2]^{1/2} \\ &\leq \tilde{C}_{\text{app}} \max(\sigma_d^2, L) \frac{(\log(n)+1)^{1/2}\sqrt{N}}{\sqrt{n}} + \frac{\tilde{C}_{\text{app}} \|\Phi\|_{L^1(\mathbb{R}^d)} (\nu + d)^{k+3}}{\sqrt{N}} \\ &\leq C_0 d^{\mathfrak{p}} \left(\frac{(\log(n)+1)^{1/2}\sqrt{N}}{\sqrt{n}} + \frac{1}{\sqrt{N}} \right) \end{aligned} \tag{100}$$

with $C_0 = \tilde{C}_{\text{app}} \max(\max(c, L), c(2\nu)^{k+3})$ and $\mathfrak{p} = p + k + 3$. This proves (i), since k and $\tilde{C}_{\text{app}}$ in Theorem 17 do not depend on d, n or N.

Proof of (ii): Let $k \in \mathbb{N}$ and $C_{\text{app}}, C_{\text{wgt}} > 0$ be as in Theorem 7, choose $\underline{p} = 2k + \frac{1}{2} + p$, $\underline{c} = C_{\text{wgt}} c (2\nu)^{2k+\frac{1}{2}}$ and let $\overline{p} > \underline{p}$, $\overline{c} > \underline{c}$. Then $\lambda \in \frac{1}{\sqrt{N}}[\underline{c} d^{\underline{p}}, \overline{c} d^{\overline{p}}]$ satisfies $\frac{1}{\sqrt{N}} C_{\text{wgt}} \|\Phi\|_{L^1(\mathbb{R}^d)} (\nu + d)^{2k+\frac{1}{2}} \leq \lambda$ and hence Theorem 19 shows that there exists $C_{\text{est}} > 0$ such that

$$\mathbb{E}[|u_d(T, \exp(\bar{X}^d)) - H_{\widehat{W}_\lambda^{d,N,n}}^{A^{d,N},B^N}(\bar{X}^d)|^2]^{1/2} \leq \frac{C_{\text{app}} \|\Phi\|_{L^1(\mathbb{R}^d)} (\nu + d)^{k+3}}{\sqrt{N}} + \frac{C_{\text{est}} d^{\overline{p}+1}}{n^{\frac{1}{4}}}. \tag{101}$$

From the proof of Theorem 19 (with $C_{\text{lam}} = \overline{c}$ here) the constant C_{est} is given by

$$C_{\text{est}}^2 = 8\overline{c}^2 \left(\frac{\nu M^2}{\nu - 2} + \mathbb{E}[|B_1|^2]\right) + 2^{3+\frac{1}{2}} \overline{c} \left(\frac{\nu M^2}{\nu - 2} \mathbb{E}[(Y_1^d)^2] + \mathbb{E}[|B_1|^2] \mathbb{E}[(Y_1^d)^2]\right)^{1/2} + 4\mathbb{E}[(Y_1^d)^4]^{1/2}$$

and hence $C_{\text{est}} \leq d^{\frac{p}{4}}\tilde{C}_{\text{est}}$ with $\tilde{C}^2_{\text{est}} = 8\overline{c}^2(\frac{\nu M^2}{\nu-2}+\mathbb{E}[|B_1|^2])+2^{3+\frac{1}{2}}\overline{c}c^{\frac{1}{4}}(\frac{\nu M^2}{\nu-2}+\mathbb{E}[|B_1|^2])^{1/2}+4c^{\frac{1}{2}}$. Thus, (101) yields

$$\mathbb{E}[|u_d(T,\exp(\bar{X}^d)) - H^{A^{d,N},B^N}_{\widehat{W}^{d,N,n}_\lambda}(\bar{X}^d)|^2]^{1/2} \leq C_0 d^{\mathfrak{p}}\left(\frac{1}{\sqrt{N}}+\frac{1}{n^{\frac{1}{4}}}\right) \tag{102}$$

with $C_0 = \max(C_{\text{app}}c(2\nu)^{k+3}, \tilde{C}_{\text{est}})$ and $\mathfrak{p} = \max(p+k+3, \overline{p}+1+\frac{p}{4})$. As shown in the above results (and visible from the explicit expressions available for these constants) neither $k \in \mathbb{N}$ nor the constants $C_{\text{app}}, C_{\text{wgt}} > 0$ depend on d, n or N. Hence, the constants $C_0, \mathfrak{p}$ do not depend on d, N, n or λ. This proves (ii).

Proof of (iii): Let $k \in \mathbb{N}$, $C_{\text{app}}, C_{\text{wgt}}$, $\underline{p}$, $\underline{c}, C_0, \mathfrak{p} > 0$ be as in the proof of (ii) and let $\overline{p} > \underline{p}$, $\overline{c} > \underline{c}$. Applying Proposition 21 with $C_{\text{lam}} = \overline{c}$ and using the estimate provided in the proof of (ii) (see (101) and (102)) for the first two terms in (76) we obtain that there exists $C_{\text{opt}} > 0$ such that

$$\mathbb{E}[|u_d(T,\exp(\bar{X}^d)) - H^{A^{d,N},B^N}_{W^{d,N,n}_{\mathcal{T}}}(\bar{X}^d)|^2]^{1/2} \leq C_0 d^{\mathfrak{p}}\left(\frac{1}{N^{\frac{1}{2}}}+\frac{1}{n^{\frac{1}{4}}}\right) + \frac{C_{\text{opt}}d^{\overline{p}+2}N(2+\log(\mathcal{T}))^{\frac{1}{2}}}{\mathcal{T}^{\frac{1}{4}}}. \tag{103}$$

The constant C_{opt} was given explicitly in the proof and we deduce that $C_{\text{opt}} \leq d^{p/4}\tilde{C}_{\text{opt}}$ with $\tilde{C}^2_{\text{opt}} = 4\max(\frac{4}{\eta_0}, 96\eta_0)\max(2, 3M^4\nu^2/[(\nu-2)(\nu-4)]+\mathbb{E}[|B_1|^4])\max(\overline{c}^2, c^{1/2})$. Combining this with (103) proves (98) with $C_1 = \max(C_0, \tilde{C}_{\text{opt}})$, $\mathfrak{q} = \max(\mathfrak{p}, \overline{p}+2+\frac{p}{4})$. By the same reasoning as above $C_1, \mathfrak{q}$ do not depend on $d, N, n, \mathcal{T}$ or λ. This proves (iii). ■

5.3 Discussion and Comparison to Deterministic Neural Networks

Let us discuss the relationship between (shallow) deterministic, fully trainable neural networks and random feature neural networks in the context of Black-Scholes PDEs. As described above, a random neural network approximation uses the function (91) with randomly generated hidden weights A^d_i, B_i and only trains the readout weights W_i, $i = 1, \dots, N$. In contrast, a standard (deterministic) neural network with a single hidden layer is a function

$$H^\theta(x) = \sum_{i=1}^N W_i \varrho(a^d_i \cdot x + b_i), \quad x \in \mathbb{R}^d, \tag{104}$$

with trainable parameters $a^d_i \in \mathbb{R}^d$, $b_i, W_i \in \mathbb{R}$, $i = 1, \dots, N$, and where we denote by $\theta = (a^d_1, \dots, a^d_N, b_1, \dots, b_N, W_1, \dots, W_N)$ the trainable parameters. Let us now compare the random feature neural network approximation and learning error bounds in Theorem 13 and Corollary 27 to existing results for (deterministic) neural networks in the context of Black-Scholes PDEs. Firstly – with the exception of Elbrächter et al. (2022) and Gonon and Schwab (2023) – existing results that prove that (deep) neural networks overcome the curse of dimensionality when approximating certain Kolmogorov PDEs (see, for instance, Grohs et al. 2023, Elbrächter et al. 2022, Berner et al. 2020, Reisinger and Zhang 2020, Gonon and Schwab 2023 and the references therein) are concerned with L^p-error bounds (for $p \in [1, \infty)$), whereas Theorem 13 provides an L^∞-error bound. Secondly, when specialized to the Black-Scholes model these papers impose a condition on boundedness or at most

polynomial growth of suitable norms of the covariance Σ^d and the drift γ^d as a function of d. In contrast, Theorem 13 and Corollary 27 impose a *non-degeneracy* condition on Σ^d. The reason for the different type of condition is that Grohs et al. (2023), Berner et al. (2020), Reisinger and Zhang (2020), Gonon and Schwab (2023) use "artificial" Monte Carlo samples to construct a neural network approximation of $u_d(T,\cdot)$, whereas Theorem 13 and Corollary 27 exploit *smoothness* of $u_d(T,\cdot)$. Thirdly, for the same reason also the condition on the payoff φ_d is different: here an integrability condition is imposed, whereas in the cited papers it is assumed that φ_d can be "approximated well" by a (deep) neural network. Finally, we comment on the case in which all conditions are satisfied and so both the approximation results for (deterministic) neural networks and those for random feature networks can be applied. In general the approximating neural networks constructed in Grohs et al. (2023), Berner et al. (2020), Reisinger and Zhang (2020), Gonon and Schwab (2023) are *deep* in the sense that they may contain more than one hidden layer, but in some situations (when the payoff is given *exactly* by a shallow neural network (104)) the approximating neural network turns out to be again a shallow neural network of type (104). Assuming the conditions on φ_d, Σ^d, γ^d are all satisfied we now obtain the same rate of decay $N^{-\frac{1}{2}}$ using a (deterministic) network (104) and a random neural network (91). This follows, e.g., from Remark 5.3 in Gonon and Schwab (2023) and Theorem 13. Alternatively, under certain hypotheses on φ_d the same approximation rate by (deterministic) shallow neural networks is also asserted by Barron (1993), see Gonon and Schwab (2021, Proposition 5.6). However, all these results for (deterministic) networks do not cover the full learning error and thus in this case random neural networks have the useful advantage that bounds on the full learning error (approximation, generalization and optimization) are available, see Corollary 27.

On the other hand, let us briefly discuss deterministic shallow neural networks from a practical perspective. Typically, finding θ that (approximately) minimizes the empirical risk $\mathcal{R}_n(H^\theta) = \frac{1}{n}\sum_{i=1}^n (H^\theta(X_i^d) - Y_i^d)^2$ is computationally more demanding than just optimizing over W in the random features case, since $\theta \mapsto \mathcal{R}_n(H^\theta)$ does not reduce to a (constrained) regression. Instead an (approximate) minimizer of $\theta \mapsto \mathcal{R}_n(H^\theta)$ is typically computed by using the stochastic gradient descent algorithm or a variant thereof (such as *Adam* introduced in Kingma and Ba 2015) and the required gradients are computed by backpropagation. Training may require tuning of the learning rate of the stochastic gradient descent algorithm and more importantly, no theoretical convergence guarantees are available in this case. Thus, from a practical implementation perspective, using random feature neural networks is advantageous in this situation.

5.4 Numerical Examples

In this section we consider numerical examples in which the solution $u_d(T,\cdot)$ to (89) is learnt from noisy observations. We carry out the same experiment for different configurations of parameters and initial conditions. We start by describing in detail the setup in a first example. We fix d, T and generate n training data points $(X_1^d, Y_1^d), \dots, (X_n^d, Y_n^d)$ for our experiment. The goal is then to learn $u_d(T,\cdot)$ based only on these data points, i.e. without using any knowledge about the underlying PDE or its parameters. This is achieved by

employing neural networks with randomly generated hidden weights, as explained in detail in Section 5.1.

For the unknown PDE we choose the pricing PDE for a max-call option in a d-dimensional Black-Scholes model with equal correlations among the assets. Thus, we fix $d = 50$, choose $\varphi_d(s) = \max(\max(s_1, \dots, s_d) - K, 0)$ as initial value for the PDE and let Σ^d be given for $i, j = 1, \dots, d$ by $\Sigma^d_{i,j} = \sigma^2\rho$ for $i \neq j$ and $\Sigma^d_{i,i} = \sigma^2$. Furthermore, $\tilde{\gamma}^d = 0$, $\nu^d_{\mathrm{L}} = 0$ and the parameter values are chosen as $\sigma = 0.2$, $\rho = 0.2$, $T = 1$. The strike K is chosen as $K = 1$ (which corresponds to expressing prices in units of the "actual" strike). From the solution $u_d(T, \cdot)$ with $K = 1$ on $\mathcal{D}^d$ one can also directly obtain the solution $\tilde{u}_d(T, \cdot)$ for other values of K (e.g. $K = 100$) on the set $\{K\exp(x) \,|\, x \in [-M, M]^d\}$ by using $\tilde{u}_d(T, s) = Ku_d(T, s/K)$. For our experiment we now select $M = 1$ and generate the i-th data point as follows: we randomly uniformly sample X^d_i on $[-1, 1]^d$ and then use a Monte Carlo simulation with $5 \cdot 10^6$ sample paths to calculate an approximate value of $u_d(T, \exp(X^d_i))$. Y^d_i is then defined as this approximate value and corresponds to a noisy observation of $u_d(T, \cdot)$ at $\exp(X^d_i)$. By using this procedure for $i = 1, \dots, n$ we generate $n = 5 \cdot 10^6$ data points (the training data).

The goal is now to learn the solution $u_d(T, \cdot)$ to (89) based only on these (noisy) observations. To achieve this we use random neural networks as described in Section 5.1. We consider different choices for the number of hidden nodes N. For the weight distributions we choose $A^d_1 \sim t_5(0, \mathbb{1}_d)$ and let B_1 have a Student's t-distribution with 2 degrees of freedom (i.e. $\nu = 5$ and π_{b} is the density of a t-distribution with 2 degrees of freedom). Unconstrained regression is employed to fit the output weights (see (93)), resulting in an output weight vector $\widehat{W}^{d,N,n}$ and a random neural network approximation $H^{A^{d,N},B^N}_{\widehat{W}^{d,N,n}}$ (see (91)) to $u_d(T, \exp(\cdot))$.

Then we generate $n_{\mathrm{test}} = 5 \cdot 10^5$ test samples $(\bar{X}^d_1, \bar{Y}^d_1), \dots, (\bar{X}^d_{n_{\mathrm{test}}}, \bar{Y}^d_{n_{\mathrm{test}}})$ according to the same procedure that we used for the training data above. Based on these training data points we calculate the squared error $\hat{e}^2 = \frac{1}{n_{\mathrm{test}}} \sum_{i=1}^{n_{\mathrm{test}}} (\bar{Y}^d_i - H^{A^{d,N},B^N}_{\widehat{W}^{d,N,n}}(\bar{X}^d_i))^2$. The error $\hat{e}$ is an estimate of the prediction error (see (92), (96) and recall that the Monte Carlo price $\bar{Y}^d_i$ is an unbiased estimate of $u_d(T, \exp(\bar{X}^d_i)))$ conditional on $A^{d,N}, B^N$. The (unconditional) prediction error $\bar{e}^2$ is then estimated by generating $\kappa = 50$ independent realizations of $A^{d,N}, B^N$, fitting $\widehat{W}^{d,N,n}$, calculating $\hat{e}^2$ for each of them and averaging $\hat{e}^2$ over the κ realizations. Figure 1 displays $\hat{e} = \hat{e}(N)$ for different choices of the number of hidden nodes, namely, $N \in \{1\} \cup \{10, 20, \dots, 190\}$ and a realization of $A^{d,N}, B^N$. The figure also displays the function $x \mapsto \frac{\hat{e}_0}{\sqrt{x}}$, where $\hat{e}_0$ is chosen as $\hat{e}(1)$. Figure 2 shows the analogous plot for the estimated (unconditional) prediction error $\bar{e}$ and the function $x \mapsto \frac{\bar{e}_0}{\sqrt{x}}$ with $\bar{e}_0 = \bar{e}(1)$.

The theoretical results from Corollary 27 show that, for n large, the theoretical prediction error decays at least as $1/\sqrt{N}$ when N increases. The numerical results here reproduce this behaviour for the (conditional and unconditional) estimated prediction errors $\hat{e}(N)$ and $\bar{e}(N)$. This can be seen from Figures 1 and 2, where the estimated errors $\hat{e}(N)$ and $\bar{e}(N)$ match closely the functions $x \mapsto \frac{\hat{e}_0}{\sqrt{x}}$ and $x \mapsto \frac{\bar{e}_0}{\sqrt{x}}$, respectively. To examine the rate of decay more precisely, we also generate a log-log plot in which $(\log(N), \log(\bar{e}(N)))$ is shown for the different choices of N above. We use linear regression to fit an affine function to these points. Since the theoretical prediction error is bounded by a constant times $\frac{1}{\sqrt{N}}$, we expect that the slope of the affine function should be close to -0.5. In fact, since the bound

may not necessarily be sharp, the slope could also be smaller than -0.5 or the behaviour could only be observed when looking at sufficiently large N. Figure 3 displays the log-log plot of the error, the regression line and a line with slope -0.5. The slope of the regression line is -0.524, confirming the expected behaviour.

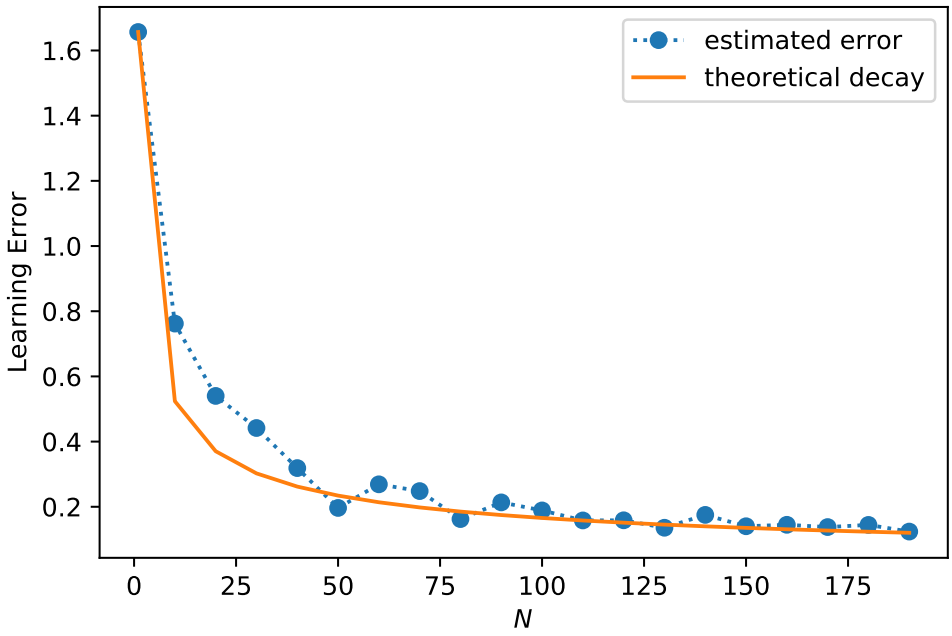

Figure 1: Plot of the estimated learning error – for a realization of the hidden weights and biases – committed when a random neural network with N hidden nodes is used to learn a 50-dimensional Black-Scholes PDE from observations. The dots show the estimated learning error $\hat{e}(N)$ for different values of N, the line shows the decay implied by the theoretical results $\frac{\hat{e}_0}{\sqrt{N}}$ (with $\hat{e}_0$ chosen as $\hat{e}(1)$).

This numerical experiment also indicates that the integrability and smoothness assumptions in Corollary 27 can potentially be relaxed. More specifically, the payoff φ_d considered in the example here does not satisfy the hypothesis $\varphi_d \circ \exp \in L^1(\mathbb{R}^d)$ and for the chosen parameters the matrix Σ^d does not satisfy (95), since the smallest eigenvalue of $\frac{1}{2}\Sigma^d$ is smaller than $\frac{1}{2^{3/2}T\pi}$ and hence any eigenvector ξ of $\frac{1}{2}\Sigma^d$ corresponding to this eigenvalue satisfies $\frac{1}{2}\xi \cdot \Sigma^d \xi < C\|\xi\|^2$ for any $C > \frac{1}{2^{3/2}T\pi}$. Nevertheless, the numerical results suggest that Corollary 27(i) is still valid in this situation. While Theorem 1 may be used to establish the $N^{-1/2}$-decay in N also without the hypotheses $\varphi_d \circ \exp \in L^1(\mathbb{R}^d)$ and (95), these hypotheses were needed in the proof of Theorem 7 (and propagate to Corollary 27) in order to guarantee that the constant in the error bound does not grow exponentially in d. The numerical experiment and the choice $d = 50$ indicates non-exponential constants also here

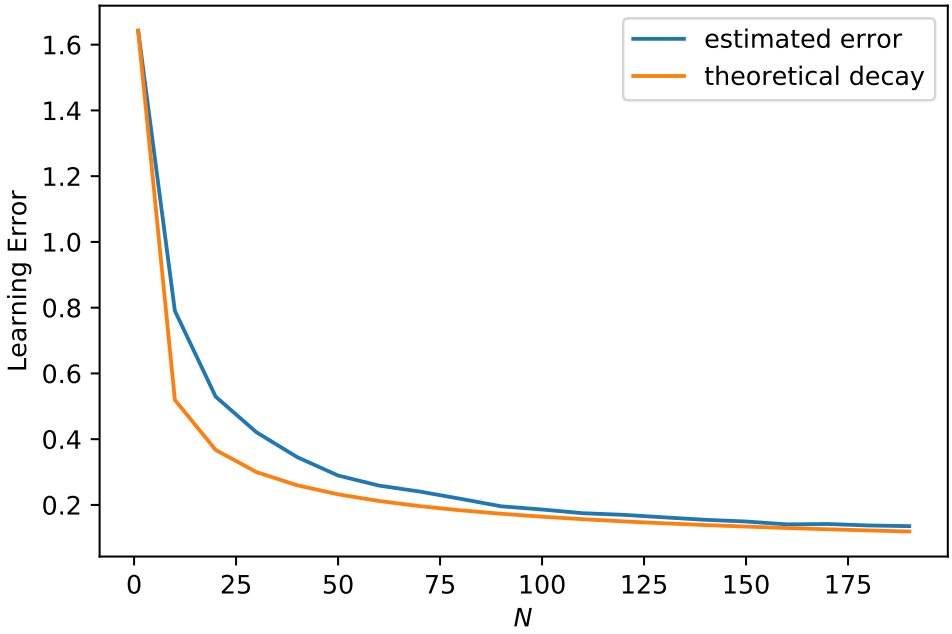

Figure 2: Plot of the estimated (unconditional) learning error committed when a random neural network with N hidden nodes is used to learn a 50-dimensional Black-Scholes PDE from observations. The blue line shows the estimated learning error $\bar{e}(N)$ for different values of N, the orange line shows the decay implied by the theoretical results $\frac{\bar{e}_0}{\sqrt{N}}$ (with $\bar{e}_0$ chosen as $\bar{e}(1)$).

and hence it may be possible to relax these assumptions by taking a different approach than the one that was used in the proof of Theorem 7.

Variations We now repeat the above experiment in different other examples and estimate in each case the rate of decay by performing a linear regression of the errors on a logarithmic scale, as described above. As training data we use $n = 2 \cdot 10^6$ data points, each of which is generated using $2 \cdot 10^6$ Monte Carlo samples. We consider several variations of the above setting.

First, we look at different payoffs given by a basket call option $\varphi_d^{(1)}(s) = \max(\frac{1}{d}\sum_{i=1}^d s_i - K, 0)$ or a put on min option $\varphi_d^{(2)}(s) = \max(K - \min(s_1, \dots, s_d), 0)$ as initial condition for the PDE. Figure 4 shows the log-log plot of the errors for $\varphi_d^{(1)}$. We observe that the expected rate of decay indeed occurs, but only for $N \geq 20$. To estimate the rate of decay we may thus omit the first two data points and obtain an estimated slope of -0.501. For $\varphi_d^{(2)}$ we observe an even faster decay and estimate a rate of -0.58. Both results again confirm the expected behaviour.

Next, we vary the number of underlying assets, i.e., the dimension d of the domain of the PDE. We consider $d \in \{1, 5, 25, 50, 200, 500\}$ and display the estimated rates in Table 1. We also indicate the range of N considered, since – as in Figure 4 above and not contradicting

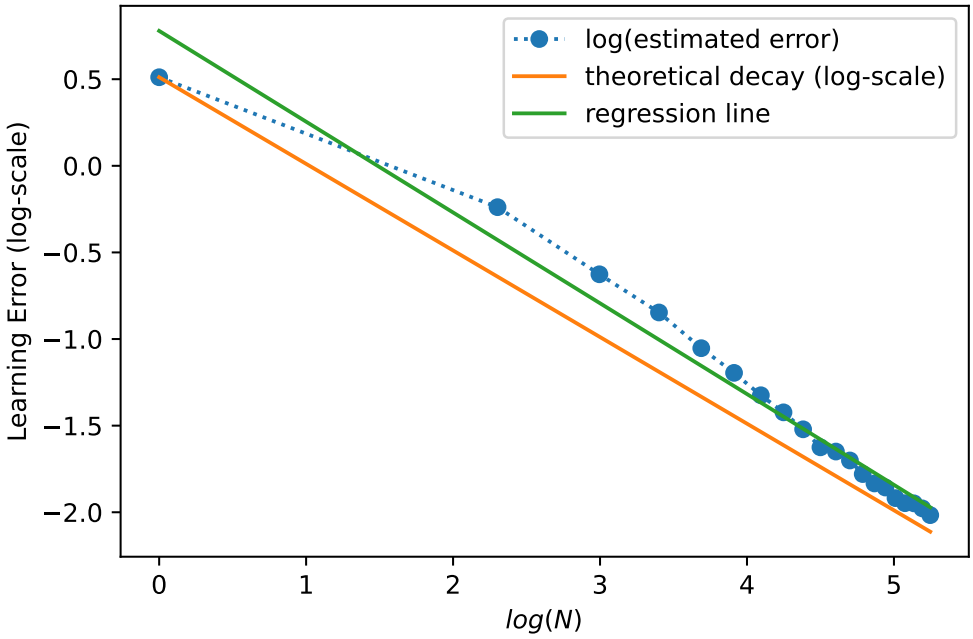

Figure 3: Log-log plot of the estimated learning error committed when a random neural network with N hidden nodes is used to learn a 50-dimensional Black-Scholes PDE from observations. The dots show $(\log(N), \log(\bar{e}(N)))$ for different values of N. The slopes of the lines correspond to the rate of decay estimated by regression (green) and expected based on the theoretical results (orange).

the non-necessarily sharp bound – for larger d the rate of convergence is only observed for N sufficiently large. For example, for $d = 500$ using $N \in \{1\} \cup \{10, 20, \ldots, 190\}$ yields the rate -0.319, whereas for $N \in \{410, 420, \ldots, 490\}$ we again observe the expected behaviour (see Figure 5 and Table 1). Thus, Table 1 shows that also for different choices of dimension d the experimental results are in line with theoretical results.

In a next experiment, we vary some further model parameters and consider different volatilities $\sigma \in \{0.01, 0.2, 0.5\}$ and correlations $\rho \in \{-0.2, 0.2\}$. The results are reported in Table 2. In each case we observe a very similar behaviour with rates smaller than -0.5.

Finally, we also consider instead of the Black-Scholes model a multivariate Merton jump diffusion model with various parameter specifications. This is an extension of the Black-Scholes model with an added independent jump process. Jump sizes are drawn independently from a $\mathcal{N}(0, \tilde{\Sigma}^d)$-distribution and jumps occur at the jump times of a Poisson process with intensity $\lambda > 0$. This corresponds to $\nu_{\mathrm{L}}^d = \lambda \nu_0$ with ν_0 a Gaussian measure with mean 0 and covariance $\tilde{\Sigma}^d$. We refer, e.g., to Eberlein and Kallsen (2019) for further details. For our experiments we let $\tilde{\Sigma}^d$ be given for $i, j = 1, \ldots, d$ by $\tilde{\Sigma}^d_{i,j} = \tilde{\sigma}^2 \tilde{\rho}$ for $i \neq j$ and $\tilde{\Sigma}^d_{i,i} = \tilde{\sigma}^2$ and consider different configurations for $\sigma, \tilde{\sigma}, \tilde{\rho}, \lambda$. The results are reported in Table 3. In

each case we observe estimated rates smaller than -0.5, which is again in line with the expected behaviour.

Table 1: Estimated rate of decay of the (unconditional) learning error in a d-dimensional Black-Scholes model with payoff $\varphi_d^{(1)}$ for varying choices of d. The rates were estimated based on $N \in \{\max(\underline{N}, 1), \underline{N} + 10, \underline{N} + 20, \dots, \overline{N}\}$.

d	1	5	25	50	200	500
Estimated rate	-1.330	-0.622	-0.562	-0.500	-0.496	-0.468
$(\underline{N}, \overline{N})$	$(0, 190)$	$(0, 190)$	$(20, 190)$	$(20, 190)$	$(200, 290)$	$(410, 490)$

Table 2: Estimated rate of decay of the (unconditional) learning error in a 50-dimensional Black-Scholes model for varying parameter choices. The rates were estimated based on $N \in \{20, 30, \dots, 190\}$.

(σ, ρ)	$(0.2, 0.2)$	$(0.01, 0.2)$	$(0.5, 0.2)$	$(0.2, -0.2)$
Estimated rate	-0.500	-0.552	-0.548	-0.519

Table 3: Estimated rate of decay of the (unconditional) learning error in a 50-dimensional Merton model for varying parameter choices. The rates were estimated based on $N \in \{20, 30, \dots, 190\}$.

$(\sigma, \tilde{\sigma}, \tilde{\rho}, \lambda)$	$(0.2, 0.1, -0.1, 1)$	$(0.2, 0.2, -0.2, 2)$	$(0.2, 0.5, 0.2, 5)$	$(0.01, 0.5, 0.2, 5)$
Estimated rate	-0.534	-0.526	-0.509	-0.514

Acknowledgments

The author would like to acknowledge support for this project by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 464123384.

References

Charalambos D. Aliprantis and Kim C. Border. *Infinite dimensional analysis*. Springer, Berlin, 2006.

Herbert Amann and Joachim Escher. *Analysis. III*. Birkhäuser, Basel, 2009.

David Applebaum. *Lévy processes and stochastic calculus*, volume 116 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, second edition, 2009.

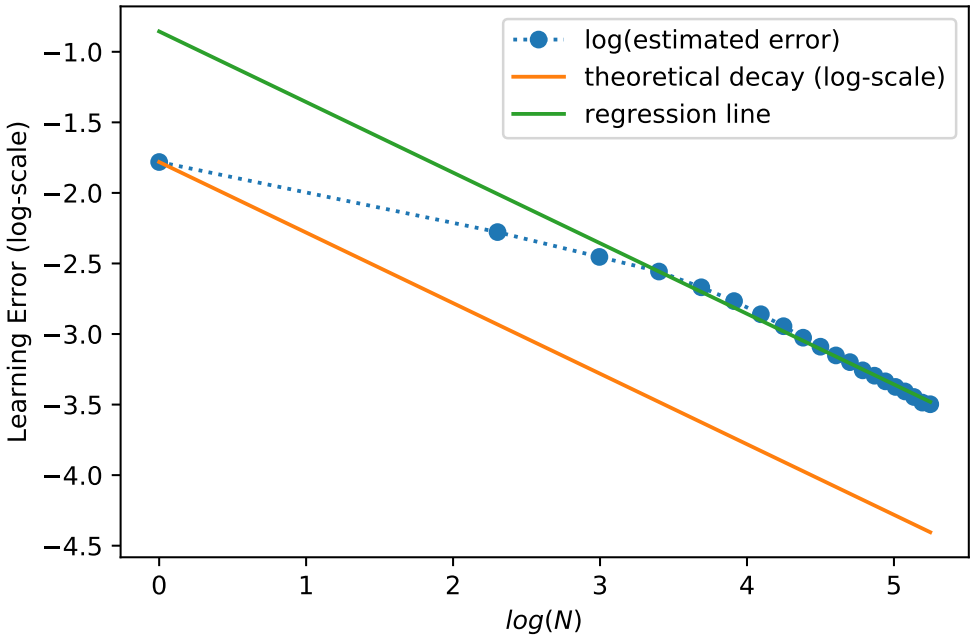

Figure 4: Log-log plot of the estimated learning error committed when a random neural network with N hidden nodes is used to learn a 50-dimensional Black-Scholes PDE from observations. In this case the payoff is given by a basket call option $\varphi_d^{(1)}(s) = \max(\frac{1}{d}\sum_{i=1}^d s_i - K, 0)$. The dots show $(\log(N), \log(\bar{e}(N)))$ for different values of N. The slopes of the lines correspond to the rate of decay estimated by regression (green) and expected based on the theoretical results (orange).

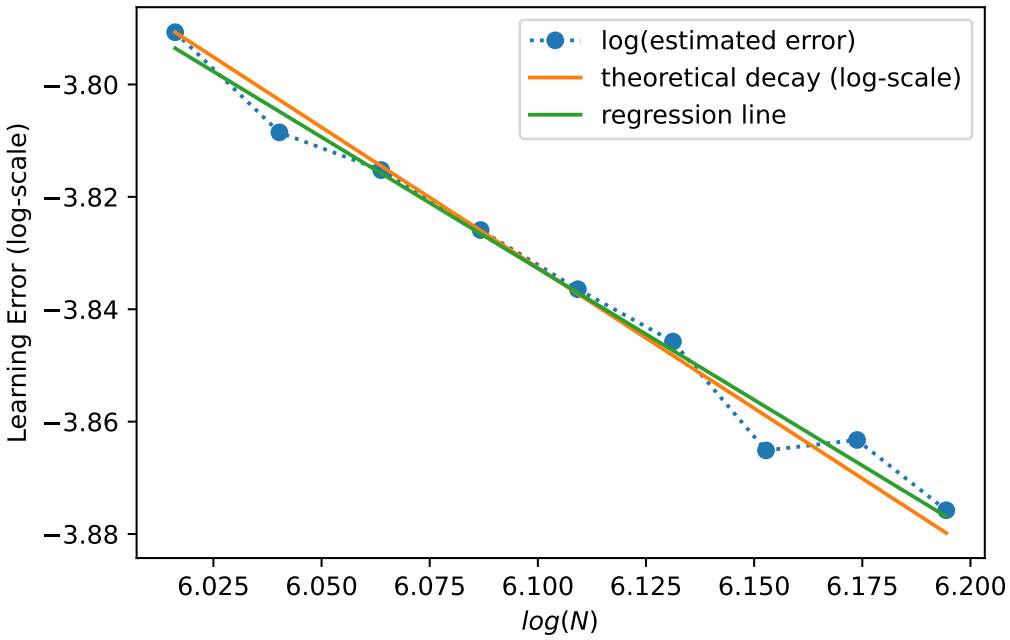

Figure 5: Log-log plot of the estimated learning error committed when a random neural network with N hidden nodes is used to learn a 500-dimensional Black-Scholes PDE from observations. The dots show $(\log(N), \log(\bar{e}(N)))$ for different values of N. The slopes of the lines correspond to the rate of decay estimated by regression (green) and expected based on the theoretical results (orange).

Guy Barles, Rainer Buckdahn, and Etienne Pardoux. Backward stochastic differential equations and integral-partial differential equations. *Stochastics Stochastics Rep.*, 60(1-2):57–83, 1997.

Andrew R. Barron. Neural net approximation. In *Yale Workshop on Adaptive and Learning Systems*, volume 1, pages 69–72, 1992.

Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inform. Theory*, 39(3):930–945, 1993.

Andrew R. Barron. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14:115–133, 1994.

Andrew R. Barron and Jason M. Klusowski. Approximation and estimation for high-dimensional deep learning networks. *Preprint, arXiv 1809.03090*, 2018.

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(3):463–482, 2003.

Christian Beck, Martin Hutzenthaler, Arnulf Jentzen, and Benno Kuckuck. An overview on deep learning-based approximation methods for partial differential equations. *Discrete and Continuous Dynamical Systems - B*, 28(6):3697–3746, 2023.

Julius Berner, Philipp Grohs, and Arnulf Jentzen. Analysis of the generalization error: empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. *SIAM J. Math. Data Sci.*, 2(3):631–657, 2020.

Julius Berner, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. The modern mathematics of deep learning. *Preprint, arXiv 2105.04026*, 2021.

Stephane Boucheron, Gabor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. OUP Oxford, 2013.

Hans Buehler, Lukas Gonon, Josef Teichmann, and Ben Wood. Deep hedging. *Quant. Finance*, 19(8):1271–1291, 2019.

Andrei Caragea, Philipp Petersen, and Felix Voigtlaender. Neural network approximation and estimation of classifiers with classification boundary in a Barron class. *The Annals of Applied Probability*, 33(4):3039 – 3079, 2023.

René Carmona and Mathieu Laurière. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games i: The ergodic case. *SIAM Journal on Numerical Analysis*, 59(3):1455–1485, 2021.

Peter Carr and Dilip B. Madan. Option valuation using the fast fourier transform. *Journal of Computational Finance*, 2:61–73, 1999.

Luigi Carratino, Alessandro Rudi, and Lorenzo Rosasco. Learning with sgd and random features. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Rama Cont and Peter Tankov. *Financial Modelling with Jump Processes*. Chapman & Hall/CRC, 2004.

Rama Cont and Ekaterina Voltchkova. Integro-differential equations for option prices in exponential lévy models. *Finance and Stochastics*, 9:299–325, 2005.

Rama Cont and Ekaterina Voltchkova. A Finite Difference Scheme for Option Pricing in Jump Diffusion and Exponential Lévy Models. *SIAM Journal on Numerical Analysis*, 43 (4):1596–1626, 2006.

Christa Cuchiero, Wahid Khosrawi, and Josef Teichmann. A generative adversarial network approach to calibration of local stochastic volatility models. *Risks*, 8(4):101, 2020.

Weinan E and Stephan Wojtowytsch. On the banach spaces associated with multi-layer relu networks: Function representation, approximation theory and gradient descent dynamics. *CSIAM Transactions on Applied Mathematics*, 1(3):387–440, 2020.

Weinan E, Jiequn Han, and Arnulf Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017.

Weinan E, Chao Ma, and Lei Wu. A priori estimates of the population risk for two-layer neural networks. *Commun. Math. Sci.*, 17(5):1407–1425, 2019.

Weinan E, Chao Ma, Stephan Wojtowytsch, and Lei Wu. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. *Preprint, arXiv 2009.10713*, 2020.

Ernst Eberlein and Jan Kallsen. *Mathematical finance*. Springer Finance. Springer, Cham, 2019.

Dennis Elbrächter, Philipp Grohs, Arnulf Jentzen, and Christoph Schwab. DNN expression rate analysis of high-dimensional PDEs: application to option pricing. *Constr. Approx.*, 55(1):3–71, 2022.

Walter Farkas, Nils Reich, and Christoph Schwab. Anisotropic stable Lévy copula processes—analytical and numerical aspects. *Math. Models Methods Appl. Sci.*, 17(9): 1405–1443, 2007.

Maximilien Germain, Huyên Pham, and Xavier Warin. Neural networks-based algorithms for stochastic control and pdes in finance. *Preprint, arXiv 2101.08068*, 2021.

Kathrin Glau. Classification of Lévy processes with parabolic Kolmogorov backward equations. *Theory Probab. Appl.*, 60(3):383–406, 2016.

Lukas Gonon and Christoph Schwab. Deep ReLU network expression rates for option prices in high-dimensional, exponential Lévy models. *Finance Stoch.*, 25(4):615–657, 2021.

Lukas Gonon and Christoph Schwab. Deep relu neural network approximation for stochastic differential equations with jumps. *Anal. Appl. (Singap.)*, 21(01):1–47, 2023.

Lukas Gonon, Philipp Grohs, Arnulf Jentzen, David Kofler, and David Šiška. Uniform error estimates for artificial neural network approximations for heat equations. *IMA J. Numer. Anal.*, 42(3):1991–2054, 2021.

Lukas Gonon, Lyudmila Grigoryeva, and Juan-Pablo Ortega. Approximation bounds for random neural networks and reservoir systems. *Ann. Appl. Probab.*, 33:28–69, 2023a.

Lukas Gonon, Lyudmila Grigoryeva, and Juan-Pablo Ortega. Infinite-dimensional reservoir computing. *Preprint, arXiv 2304.00490*, 2023b.

Philipp Grohs, Fabian Hornung, Arnulf Jentzen, and Philippe von Wurstemberger. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. *Mem. Amer. Math. Soc.*, 284:1–106, 2023.

László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A distribution-free theory of nonparametric regression.* Springer Series in Statistics. Springer-Verlag, New York, 2002.

Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34): 8505–8510, 2018.

Norbert Hilber, Nils Reich, Christoph Schwab, and Christoph Winter. Numerical methods for Lévy processes. *Finance and Stochastics*, 13:471–500, 2009.

Guang-Bin Huang, Lei Chen, and Chee-Kheong Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *Trans. Neur. Netw.*, 17(4):879–892, July 2006.

Côme Huré, Huyên Pham, and Xavier Warin. Deep backward schemes for high-dimensional nonlinear PDEs. *Math. Comp.*, 89(324):1547–1579, 2020.

Martin Hutzenthaler, Arnulf Jentzen, Thomas Kruse, Tuan Anh Nguyen, and Philippe von Wurstemberger. Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations. *Proc. A.*, 476(2244):630–654, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2015. 3rd International Conference for Learning Representations, San Diego, 2015.

Jason M. Klusowski and Andrew R. Barron. Approximation by combinations of ReLU and squared ReLU ridge functions with ℓ^1 and ℓ^0 controls. *IEEE Trans. Inform. Theory*, 64 (12):7649–7656, 2018.

Gitta Kutyniok, Philipp Petersen, Mones Raslan, and Reinhold Schneider. A theoretical analysis of deep neural networks and parametric PDEs. *Constr. Approx.*, 55:73–125, 2022.

Fabian Laakmann and Philipp Petersen. Efficient approximation of solutions of parametric linear transport equations by ReLU DNNs. *Adv. Comput. Math.*, 47(1):Paper No. 11, 2021.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces*. Springer Berlin Heidelberg, 2013.

Yulong Lu, Jianfeng Lu, and Min Wang. A priori generalization analysis of the deep ritz method for solving high dimensional elliptic partial differential equations. In *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 3196–3241. PMLR, 15–19 Aug 2021.

Tao Luo and Haizhao Yang. Two-layer neural networks for partial differential equations: Optimization and generalization theory. *Preprint, arXiv 2006.15733*, 2020.

V. E. Maiorov and R. Meir. On the near optimality of the stochastic approximation of smooth functions by neural networks. *Adv. Comput. Math.*, 13(1):79–103, 2000.

Ana-Maria Matache, Tobias von Petersdorff, and Christoph Schwab. Fast deterministic pricing of options on Lévy driven assets. *M2AN Math. Mod. and Num. Anal.*, 38:37–71, 2004.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4):667–766, 2022.

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Generalization error of random feature and kernel methods: Hypercontractivity and kernel matrix concentration. *Applied and Computational Harmonic Analysis*, 59:3–84, 2022.

Huyên Pham. Optimal stopping of controlled jump diffusion processes: a viscosity solution approach. *J. Math. Systems Estim. Control*, 8(1):27 pp. 1998.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2008.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems*, volume 21, pages 1313–1320, 2009.

Christoph Reisinger and Yufei Zhang. Rectified deep neural networks overcome the curse of dimensionality for nonsmooth value functions in zero-sum games of nonlinear stiff systems. *Anal. Appl. (Singap.)*, 18(6):951–999, 2020.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017.

Johannes Ruf and Weiguan Wang. Neural networks for option pricing and hedging: a literature review. *J. Comput. Finance*, 24(1):1–46, 2020.

Ken-Iti Sato. *Lévy processes and infinitely divisible distributions*. Cambridge University Press, 1999.

Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 71–79. PMLR, 17–19 Jun 2013.

Jonathan W. Siegel and Jinchao Xu. Approximation rates for neural networks with general activation functions. *Neural Networks*, 128:313–321, 2020.

Justin Sirignano and Konstantinos Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *J. Comput. Phys.*, 375:1339–1364, 2018.

Josef Stoer and Roland Bulirsch. *Introduction to numerical analysis*, volume 12 of *Texts in Applied Mathematics*. Springer-Verlag, New York, third edition, 2002.