A Priori Error Estimates For Shallow Physically Informed Neural Networks In One Dimension

Umberto Zerbinati

December 6, 2021

1 Introduction

Neural networks have been used in a large variety of applications. Lately more and more interest has focused on using neural networks to solve partial differential equations. Numerical evidence and theoretical results suggest that neural networks might be able to beat the curse of dimensionality. Different approaches have been taken into consideration to apply neural networks to solve partial differential equations, of particular interest are physically informed neural networks (PINNs) and the Finite Neuron Method (FNM). While for the FNM there is a robust error analysis, PINNs have found their way in many more practical applications, [1]. The robustness of the error analysis for the FNM comes from the connections that have been found between this method and linear finite elements. In this report a connection between LSQFEM and the PINNs is drawn, furthermore we exploit such a connection in order to develop robust error estimate for PINNs using techniques developed in the context of the FNM, [2].

2 Galerkin Least Square

Let X and Y two Hilbert spaces and consider a Fredholm operator $Q \in \mathcal{L}(X,Y)$, we will now focus our attention on the following problem,

$$Find\ u \in X\ such\ that\ Qu = F. \tag{1}$$

Definition 2.1 (Fredholm Operator)**.** *A linear and bounded operator $Q \in \mathcal{L}(X,Y)$ is called a Fredholm operator if it verifies the following conditions,*

1. *the range of the operator, $R(Q)$ is closed;*
2. *the dimension of the kernel is finite, i.e.* $\dim\Big(N(Q)\Big) < \infty$*;*
3. *the codimension of the operator is finite, i.e.* $codim\Big(R(Q)\Big) < \infty$*;*

in particular given a Fredholm operator we can define the Fredholm index of such operator as,

$$ind(Q) = dim\Big(N(Q)\Big) - codim\Big(R(Q)\Big).$$

We will work under the following assumption to simplify the discussion, $\dim\Big(N(Q)\Big) = 0$. We will focus our interest on the **residual energy functional** and we consider the following minimization principle,

$$J(u;F) = \|Qu - F\|_Y^2, \qquad u = \underset{u\in X}{\arg\min}\ J(u;F). \tag{2}$$

The above minimisation principle admit a unique minimizer, to show this we use the following lemmas.

Lemma 2.1. *Given a reflexive Banach space X and a continuous and strongly convex function,*

$$J : X \to \mathbb{R},$$

if the following conditions are satisfied,

1. $\lim_{\|x\|\to\infty} J(x) = \infty$,
2. *K is a closed convex subset of X,*

than it exist a unique element $x^* \in K$ such that,

$$J(x^*) = \min_{x\in K} J(x).$$

Lemma 2.2. *A closed subspace of an Hilbert space is a Hilbert space with respect to the inner product of the space.*

Theorem 2.1. *The minimization principle* (2) *has a unique minimizer $u^* \in X$ for any $F \in Y$.*

Proof. First we want to prove the ellipticity of the operator Q. To do this we notice that by assumption $R(Q)$ is closed in Y and therefore $\Big(R(Q), (\cdot,\cdot)_Y\Big)$ is a Hilbert space. Now we consider the restricted mapping,

$$Q : X \to R(Q),$$

such mapping is a bijection because of the assumption $\dim\Big(N(Q)\Big) = 0$ and the fact we have taken as codomain of the operator $R(Q)$. Since the mapping is a bijection one can use the bounded inverse theorem to state that $Q^{-1} : R(Q) \to X$ is a liner bounded operator and therefore,

$$\|Qu\|_Y \geq C_1\|u\|_X. \tag{3}$$

Now we proceed to prove the coercivity for $J(u;F)$, which is property 1, in the theorem statement.

$$J(u;F) = \Big(Qu - F, Qu - F\Big)_Y = (Qu, Qu)_Y - 2(Qu, F)_Y + (F, F)_Y = \|Qu\|_Y^2 + \|f\|_Y^2 - 2(Qu, f)_Y$$

using Yang inequality with $\varepsilon = 2$ and (3) we get the following inequality,

$$J(u;F) \geq \frac{1}{2}\|Qu\|_Y^2 - \|F\|_Y^2 \geq C_1\|u\|_X^2 - \|F\|_Y^2,$$

which imply $\lim_{\|x\|\to\infty} J(x) = \infty$. The only thing left to prove is that the functional $J(u;F)$ is strictly convex, to do this we consider $u, v \in X$ and $t \in [0,1]$ and we evaluate,

$$j(t) = J(tu+(1-t)v;F) = t^2(Qu-F,Qu-F)_Y + (1-t)^2(Qv-F,Qv-F)_Y + t(1-t)(Qu-F,Qv-F)_Y.$$

If the coefficient of the quadratic term is greater then zero then the functional $J(u;F)$ is convex and this is the case because,

$$(Qu-f,Qu-f)_Y - 2(Qu-F,Qv-F)_Y + (Qv-F,Qv-F)_Y \geq 0$$

where we have used Yang's inequality to obtain the last bound. In particular we notice that if $(Qu-f,Qu-f)_Y - 2(Qu-F,Qv-F)_Y + (Qv-F,Qv-F)_Y$ is null then,

$$(Qu-Qv,Qu-Qv) = 0 \Leftrightarrow Qu - Qv = \Leftrightarrow Q(u-v) = 0 \Leftrightarrow u = v,$$

where the last implication comes from the fact that we assume $\dim\Big(N(Q)\Big) = 0$. To conclude one need to apply the previous Lemma. □

In particular one can characterize the minimizer of (2) using the following proposition.

Proposition 2.1. *The minimizer u^* of (2) solves the following variational equation,*

$$\text{Find } u \in X \text{ such that } a(u,v) = G(v)\ \forall v \in X, \tag{4}$$

where $a(u,v) = \Big(Qu, Qv\Big)_Y$ and $G(v) = (f, Qv)_Y$.

Proof. We define the following function of a real variable $j(t) = J(u^* + tv; F)$ and we notice that since u^* is a minimum for $J(u;F)$ then $j'(0) = 0$. Expanding $j'(t)$ we get,

$$j'(t) = 2t(Qv, Qv)_Y + 2(Qv, Qu - F)_Y.$$

Imposing $j'(0) = 0$ for all $v \in X$ one finds (4). □

Proposition 2.2. *The solution of (4) is also the unique minimizer of (2).*

Proof. This is an application of the Hilbert projection theorem for linear subspace. □

From now on we will consider the following version of (1),

$$\begin{aligned} Q : X \to Y = A(\Omega) \times B(\partial\Omega) \\ u \mapsto \Big(\mathcal{L}u, \mathcal{B}u\Big). \end{aligned}$$

3 Physically Informed Neural Network

Definition 3.1 (Forward Neural Network)**.** *We say $\mathcal{N}^L(\mathbf{x}) : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ is a $(L-1)$ hidden layer forward neural network (FNN) with N_ℓ neurons in the $\ell - th$ layer and activation function $\sigma : \mathbb{R} \to \mathbb{R}$, if the action $\mathcal{N}^L(\mathbf{x})$ is recursively defined as,*

1. $\mathcal{N}^0(\mathbf{x}) = \mathbf{x} \in \mathbb{R}^{d_{in}}$,
2. $\forall 1 \le \ell \le L-1 \qquad N^\ell(\mathbf{x}) = \sigma(W^\ell N^{\ell-1}(\mathbf{x}) + \mathbf{b}_\ell) \in \mathbb{R}^{N_\ell}$,
3. $\mathcal{N}^L(\mathbf{x}) = W^L \mathcal{N}^{L-1}(\mathbf{x}) + \mathbf{b}^L \in \mathbb{R}^{\varkappa \approxeq \approx}$,

where $W^\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $\mathbf{b}_\ell \in \mathbb{R}^{N_\ell}$. In particular the value of the matrix W^ℓ will be called kernel parameters and the value of the vector $\mathbf{b}^\ell$ will be called bias parameters for the layer ℓ. The parameters of the network will be denoted as $\theta = \Big(\{W_\ell\}_{\ell=1}^L, \{\mathbf{b}_\ell\}_{\ell=1}^L\Big)$ and we indicate the dependence of the FNN on a particular choice of parameters $\hat{\theta}$ writing $\mathcal{N}^L_{\hat{\theta}}$.

We will focus our attention on the logistic sigmoid, i.e. $\sigma(x) = \frac{1}{1+e^{-x}}$, and the hyperbolic tangent as activation functions. When we speak about **Physically Informed Neural Networks (PINNs)**, we mean a FNN trained to solve (1) using the following algorithm:

1. Given a probability distribution of points on a set A called $\mathcal{D}(A)$ we consider two vector of realisations, i.e.
$$\omega_i \sim \mathcal{D}(\Omega) \qquad \forall 1 \le i \le N_\Omega \qquad\qquad \beta_i \sim \mathcal{D}(\partial\Omega) \qquad \forall 1 \le i \le N_{\partial\Omega}.$$
2. We consider the following loss function, i.e.
$$\mathscr{L}(\theta; \omega_i, \beta_i) = \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} \big\| \mathcal{L}\mathcal{N}^L(\omega_i) - f(\omega_i) \big\|_a^2 + \frac{\gamma}{N_{\partial\Omega}} \sum_{i=1}^{N_{\partial\Omega}} \big\| \mathcal{B}\mathcal{N}^L(\beta_i) - g(\beta_i) \big\|_b^2, \tag{5}$$
where γ is the boundary penalty parameter.

3. We train the FNN in order to find θ^* that minimize the above lost function.

It is worth it to notice that the optimization problem in step 3 of the PINNs algorithm is highly non convex with respect to θ, [3], in particular to train the PINN the state of the art would be to use gradient descent method as Adam and L-BFGS, [4],[1], even if there are evidence that second order method might be more suited to train PINNs [**?**][1]. Approximation theory and error estimates have been briefly addressed in [1], while in [5] one can find a detailed error analysis for a large class of linear parabolic differential equations. In particular as observed in [1], one might wonder if a neural network exists that can uniformly approximate a function and all its partial derivatives. The correct functional framework in which to address this problem has been identified in [6],[7],[8]. Let us consider the class of shallow FNN with activation function σ, i.e.

$$\Sigma_N^d(\sigma) = \Big\{ \sum_{i=1}^N \sigma(\mathbf{w}_i \cdot x + \mathbf{b}_i) \ : \ \mathbf{w}_i \in \mathbb{R}^d \text{ and } \mathbf{b}_i \in \mathbb{R} \Big\}$$

Theorem 3.1 (Pinkus, [9],[10])**.** *The space Σ_N^d is dense in the space of smooth functions, in particular given $f \in C^\alpha(R^d)$, a compact set $K \subset R^d$ and $\varepsilon > 0$ than it exists an $f_N \in \Sigma_N^d$ such that,*

$$\max_{\mathbf{x} \in K} |D^\alpha f(\mathbf{x}) - D^\alpha g(\mathbf{x})| \leq \varepsilon,$$

for all multi-index $\mathbf{k}$ *such that $|k| \leq \alpha$.*

This shows that with a shallow FNN one can uniformly approximate a sufficiently smooth function and all its partial derivatives. Let us now extend our question further, i.e. what are the functions that can be efficiently approximated by shallow neural networks ?

Definition 3.2. *Given a Banach space X, $\mathbb{D} \subset X$ such that $\sup_{d \in \mathbb{D}} \|d\|_X = K_{\mathbb{D}} < \infty$ we consider the closure of the convex symmetric hull of $\mathbb{D}$,*

$$B_1(\mathbb{D}) = \overline{\Big\{ \sum_{j=1}^n a_j d_j \ : \ n \in \mathbb{N}, d_j \in \mathbb{D} \text{ and } \big\|\{a_j\}_{j=1}^n\big\|_{\ell_1} \leq 1 \Big\}}.$$

Then one defines the norm , $\|\cdot\|_{\mathcal{K}_1(\mathbb{D})}$ and the space $\mathcal{K}_1(\mathbb{D})$ as:

$$\|\cdot\|_{\mathcal{K}_1(\mathbb{D})} = \inf \Big\{ c > 0 \ : \ cf \in B_1(\mathbb{D}) \Big\} \text{ and } \mathcal{K}_1(\mathbb{D}) = \Big\{ f \in X \ : \ \|f\|_{\mathcal{K}_1(\mathbb{D})} < \infty \Big\} \subset X.$$

The significance of the above space $\mathcal{K}_1(\mathbb{D})$ comes from the following result,

Theorem 3.2 (Maurey, [11],[12])**.** *Let X be a type-2 Banach space and $f \in K_1(\mathbb{D})$ then the following approximation estimate holds,*

$$\inf_{f_N \in \Sigma_{N,M}(\mathbb{D})} \|f - f_n\|_X \leq C_X K_{\mathbb{D}} \|f\|_{\mathcal{K}_1(\mathbb{D})} n^{-\frac{1}{2}},$$

where $\Sigma_{N,M}(\mathbb{D}) = \Big\{ \sum_{j=1}^N a_j d_j \ : \ d_j \in \mathbb{D} \text{ and } \big\|\{a_j\}_{j=1}^N\big\|_{\ell_1} \leq M \Big\}$. We will focus on the dictionary $\mathbb{D}_\sigma = \Big\{ \sigma(\mathbf{w}_i \cdot x + \mathbf{b}_i) \ : \ \mathbf{w}_i \in \mathbb{R}^d \text{ and } \mathbf{b}_i \in \mathbb{R} \Big\}$ for which $K_{\mathbb{D}}$ depends on σ. When σ is a bounded activation function than $\mathbb{D}_\sigma = \Big\{ \sigma(\mathbf{w}_i \cdot x + \mathbf{b}_i) \ : \ \mathbf{w}_i \in \mathbb{R}^d \text{ and } \mathbf{b}_i \in \mathbb{R} \Big\}$ then $\Sigma_{N,M}$ is uniformly bounded in $L^p(\Omega)$.

In the case of the sigmoid function, it is possible to prove that if we chose as activation function the sigmoid then $\mathcal{K}_1(\mathbb{D}_\sigma) = \mathcal{K}_1\Big(\mathbb{P}_0\Big)$, this gives us an improved approximation estimate,

$$\inf_{f_N \in \Sigma_{N,M}(\sigma)} \|f - f_N\|_X \lesssim N^{-\frac{1}{2} - \frac{1}{2d}},$$

more detail can be found in [13],[6],[14].

[1] Work in progress with KAUST Extreme Computing Research Center.

4 A Priori Error Estimate

We will perform our a priori error analysis in the one dimensional case, in particular the first thing we would like to do is to notice that if ω_i are equally spaced than the loss function (5) is the discretization a mid point quadrature of

$$J(u;F) = \|Qu - F\|_Y^2 = \int_Y \|Qu - F\|_y = \int_\Omega \|\mathcal{L}u - f\|_a + \gamma \int_{\partial\Omega} \|\mathcal{B}u - g\|_b$$

where $Q = (\mathcal{L}, \mathcal{B})$, the space $Y = A(\Omega) \times B(\partial\Omega)$ and the space A and B are equipped with the following scalar product and induced norm,

$$(u,v)_{A(\Omega)} = \int_\Omega (u(\mathbf{x}), v(\mathbf{x}))_a \, d\mathbf{x} \qquad \|u\|_{A(\Omega)} = \int_\Omega (u(\mathbf{x}), u(\mathbf{x}))_a \, d\mathbf{x} = \int_\Omega |u(\mathbf{x})|_a^2 \, d\mathbf{x},$$
$$(u,v)_{B(\partial\Omega)} = \gamma \int_{\partial\Omega} (u(\mathbf{x}), v(\mathbf{x}))_b \, d\mathbf{x} \qquad \|u\|_{B(\partial\Omega)} = \gamma \int_{\partial\Omega} (u(\mathbf{x}), u(\mathbf{x}))_b \, d\mathbf{x} = \gamma \int_{\partial\Omega} |u(\mathbf{x})|_b^2 \, d\mathbf{x}$$

Now since the mid point rule exactly integrates polynomials of order one the idea would be to use Bramble Hilbert lemma to obtain,

$$|J(u;F) - \mathscr{L}(u)| \leq C N_\Omega^{-\frac{2}{d}} \|l_u\|_{1,\infty} + C N_{\partial\Omega}^{-\frac{2}{d-1}} \|b_u\|_{1,\infty} \tag{6}$$

where $l_u(\mathbf{x}) \mapsto \|\mathcal{L}u(\mathbf{x}) - f(\mathbf{x})\|_a$ and $b_u(\mathbf{x}) \mapsto \|\mathcal{B}u(\mathbf{x}) - g(\mathbf{x})\|_b$ lives respectively in $\mathcal{W}^{1,\infty}(\Omega)$ and $\mathcal{W}^{1,\infty}(\partial\Omega)$. Now we can proceed to estimate the difference between u defined as in (2) and u_θ which is defined as

$$u_\theta = \underset{v \in \Sigma_{N,M}(D_\sigma)}{\arg\min} \mathscr{L}(v;F) \tag{7}$$

as follow,

$$\begin{aligned}
\|u_\theta - u\|_X^2 &\leq \|Q(u_\theta - u)\|_Y^2 \\
&= (Qu_\theta, Qu_\theta) - 2(Qu, Qu_\theta) + (Qu, Qu) \\
&= (Qu_\theta, Qu_\theta) - 2(F, Qu_\theta) + (F,F) - (Qu, Qu) + 2(F, Qu) - (F,F) \\
&= J(u_\theta;F) - J(u;F) \leq \mathscr{L}(u_\theta) - \mathscr{L}(u) + |J(u_\theta;F) - \mathscr{L}(u_\theta)| + |J(u;F) - \mathscr{L}(u)|
\end{aligned} \tag{8}$$

we notice last two term of the above inequality are bounded by (6), to estimate the first term in the inequality in [2] a greedy algorithm is used to minimize (7). The greedy algorithm was first proposed to minimize quadratic functional in [15] and it is studied in detail when applied to the problem here described in [16]. In particular this greedy algorithm consist in recursively constructing a sequence that converges to u_θ, as follow:

$$\begin{aligned}
u_0 &= 0, \\
g_k &= \underset{g \in \mathbb{D}_\sigma}{\arg\max} \Big\langle \mathscr{L}(u_{k-1}), g \Big\rangle \\
u_k &= (1 - s_k) u_{k-1} - M s_k g_k,
\end{aligned} \tag{9}$$

where $s_k = \min\left(1, \frac{2}{k}\right)$. The following lemma will allow us to provide an upper bound for $\mathscr{L}(u_\theta) - \mathscr{L}(u)$,

Lemma 4.1. *Let $\mathbb{D}_\sigma$ be a symmetric dictionary, such that $\underset{d \in \mathbb{D}_\sigma}{\|d\|}_X \leq C < \infty$. Furthermore we consider u_N the N-th iteration of the greedy algorithm, defined as in* (9)*, then if $\mathscr{L}$ is a K-smooth functional and the* $\arg\max$ *in* (9) *is solved up to a factor $R > 1$, i.e.*

$$\underset{g \in \mathbb{D}_\sigma}{\arg\max} \Big\langle \mathscr{L}(u_{k-1}), g_k \Big\rangle \geq \frac{1}{R} \underset{g \in \mathbb{D}_\sigma}{\arg\max} \Big\langle \mathscr{L}(u_{k-1}), g \Big\rangle,$$

then $\|u_N\|_{\mathcal{K}_1(\mathbb{D}_\sigma)} \leq M$ *and the following upper bound holds*

$$\mathscr{L}(u_N) - \inf_{\|v\|_{\mathcal{K}_1(\mathbb{D}_\sigma)} \leq \frac{M}{R}} \mathscr{L}(v) \leq 32CM^2KN^{-1}.$$

Proof. The proof of this lemma can be found in [15] and [16]. The only difference is that the minimization principle we are here considering is on $\Sigma_{N,M}$ rather then on $B_M(\mathbb{D}_\sigma)$ but this is not a concern since the above greedy algorithm (9) at each step produce a finite width neural network that correspond to the minimization principle (7). □

Theorem 4.1. *Given* $u \in \mathcal{K}_1(\mathbb{D}_\sigma)$ *defined as in* (2) *and* u_θ^N *the* N*-th iteration of* (9) *that approximate* u_θ *defined as in* (7)*, in the one dimensional setting with equally spaced* ω_i*, then we get the following a priori error estimate,*

$$\left\|u_\theta^N - u\right\|_X^2 \lesssim N_\Omega^{-2}\|l_u\|_{1,\infty} + N^{-1}.$$

provided that $l_u(\mathbf{x}) \mapsto \|\mathcal{L}u(\mathbf{x}) - f(\mathbf{x})\|_a$ *and* $b_u(\mathbf{x}) \mapsto \|\mathcal{B}u(\mathbf{x}) - g(\mathbf{x})\|_b$ *lives respectively in* $\mathcal{W}^{1,\infty}(\Omega)$ *and* $\mathcal{W}^{1,\infty}(\partial\Omega)$.

Proof. We start from (8) and notice that after using Bramble Hilbert to bound the last two terms one has,

$$\left\|u_\theta^N - u\right\|_X^2 \leq \mathscr{L}(u_\theta) - \mathscr{L}(u) + CN_\Omega^{-\frac{2}{d}}\|l_u\|_{1,\infty} + CN_{\partial\Omega}^{-\frac{2}{d-1}}\|b_u\|_{1,\infty},$$

using the previous Lemma we can also bound the first term, to obtain:

$$\left\|u_\theta^N - u\right\|_X^2 \leq 32CM^2KN^{-1} + CN_\Omega^{-\frac{2}{d}}\|l_u\|_{1,\infty} + CN_{\partial\Omega}^{-\frac{2}{d-1}}\|b_u\|_{1,\infty}.$$

It is important to notice that since $J(u;F)$ is a quadratic functional it is also K-smooth and this also imply that $\mathscr{L}$ is K-smooth. □

An other way of estimating the first quantity in (8) is to use ideas that comes from standard finite element method, in particular the notion of discrete least square minimisation principle applied to $\mathscr{L}$, but this idea will be expanded in future work.

References

[1] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. DeepXDE: a deep learning library for solving differential equations. *SIAM Rev.*, 63(1):208–228, 2021.

[2] Qingguo Hong, Jonathan W Siegel, and Jinchao Xu. A priori analysis of stable neural network solutions to numerical pdes. *arXiv preprint arXiv:2104.02903*, 2021.

[3] Avrim L. Blum and Ronald L. Rivest. Training a 3-node neural network is NP-complete. In *Machine learning: from theory to applications*, volume 661 of *Lecture Notes in Comput. Sci.*, pages 9–28. Springer, Berlin, 1993.

[4] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. USA*, 115(34):8505–8510, 2018.

[5] Tim De Ryck and Siddhartha Mishra. Error analysis for physics informed neural networks (pinns) approximating kolmogorov pdes. *arXiv preprint arXiv:2106.14473*, 2021.

[6] Jonathan W Siegel and Jinchao Xu. Sharp lower bounds on the approximation rate of shallow neural networks. *arXiv preprint arXiv:2106.14997*, 2021.

[7] E Weinan, Chao Ma, and Lei Wu. Barron spaces and the compositional function spaces for neural network models. *arXiv preprint arXiv:1906.08039*, 2019.

[8] Stephan Wojtowytsch and E Weinan. Can shallow neural networks beat the curse of dimensionality? a mean field training perspective. *IEEE Transactions on Artificial Intelligence*, 1(2):121–129, 2020.

[9] Allan Pinkus. Approximation theory of the MLP model in neural networks. In *Acta numerica, 1999*, volume 8 of *Acta Numer.*, pages 143–195. Cambridge Univ. Press, Cambridge, 1999.

[10] Allan Pinkus. *n-widths in approximation theory*, volume 7 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*. Springer-Verlag, Berlin, 1985.

[11] Bernard Maurey and Gilles Pisier. Séries de variables aléatoires vectorielles indépendantes et propriétés géométriques des espaces de banach. *Studia Mathematica*, 58(1):45–90, 1976.

[12] Ronald A. DeVore. Nonlinear approximation. In *Acta numerica, 1998*, volume 7 of *Acta Numer.*, pages 51–150. Cambridge Univ. Press, Cambridge, 1998.

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

[14] Jonathan W Siegel and Jinchao Xu. Characterization of the variation spaces corresponding to shallow neural networks. *arXiv preprint arXiv:2106.15002*, 2021.

[15] Lee K. Jones. A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *Ann. Statist.*, 20(1):608–613, 1992.

[16] Jun Wang, Chao Jin, Souhail Meftah, and Khin Mi Mi Aung. Popcorn: Paillier meets compression for efficient oblivious neural network inference. *arXiv preprint arXiv:2107.01786*, 2021.