Approximations with neural networks

Project report
February 12th, 2020

Authors :
Maéva Caillat
Stephen Jaud
Gauthier Pezzoli
Pierre-Jean Pollot

Supervisor :
Anthony Nouy

Contents

Introduction

Neural networks have been developed since 1943 thanks to the research works of *McCulloch* and *Pitts*. Nevertheless, the computational power at this time necessitated a restrained number of hidden-layers.

Nowadays, recent technologies have overcome this issue so that the discipline has become really popular in various fields, like quantum chemistry or molecule dynamics. Still, neural networks are lacking mathematical foundations and results are often exhibited by empiric ways.

Yet, recently, multiple successful applications of deep neural networks to pattern recognition problems have revived active interest in the theoretical properties of such networks.

In this project, we study results about approximations with neural networks, provided by four articles, from 1998 to 2019. The next steps will be to validate these results through experimentation, using specific functions classes and deep learning Python libraries.

Chapter 1

Nonlinear Approximation

The fundamental problem of approximation theory is to resolve a possibly complicated function, called the target function, by simpler, easier to compute functions called the approximants. The early methods utilized approximation from finite-dimensional linear spaces. It was noted shortly thereafter that there was some advantage to be gained by not limiting the approximations to come from linear spaces, and therein emerged the beginnings of nonlinear approximation. A central goal in approximation theory is to characterize functional spaces which can be approximated with a certain efficiency.

1.1 Approximation in a Hilbert space

We discuss here the case of approximation in a Hilbert space $\mathcal{H}$ with inner product $<,>$ and norm $||\ ||_{\mathcal{H}}$ and let $(\eta_k)_{k\geq 1}$ be an orthonormal basis of $\mathcal{H}$.

1.1.1 Linear Approximation

In the case of linear approximation, we use the linear space $\mathcal{H}_n = \text{span}\,\{\eta_k \,|\, 1 \leq k \leq n\}$ for approximate an element $f \in \mathcal{H}$.

Definition 1.1. Let $f \in \mathcal{H}$ and $n \in \mathbb{N}^*$. We define the *approximation error* by :

$$E_n(f)_{\mathcal{H}} = \inf_{g \in \mathcal{H}_n} ||f - g||_{\mathcal{H}} \tag{1.1}$$

Definition 1.2. Let $\alpha \in \mathbb{R}_+^*$ We define the functional space $\mathcal{A}^\alpha\left((\mathcal{H}_n)_{n\geq 1}\right)$ by

$$\mathcal{A}^\alpha\left((\mathcal{H}_n)_{n\geq 1}\right) = \left\{ f \in \mathcal{H} \;\middle|\; \sup_{n\geq 1} n^\alpha E_n(f)_{\mathcal{H}} < \infty \right\} \tag{1.2}$$

We now want to characterize the space $\mathcal{A}^\alpha((\mathcal{H}_n))$. As f is an element of a Hilbert space the best approximation to f, from $\mathcal{H}_n$ in the sense of the *approximation error* defined above, is given by the projection $P_n f = \sum_{k=1}^n f_k \eta_k$ where $f_k =< f, \eta_k >$.

Proposition 1.1. *Let $f \in \mathcal{H}$ and $\alpha \in \mathbb{R}_+^*$ then*

$$f \in \mathcal{A}^\alpha\left((\mathcal{H}_n)_{n\geq 1}\right) \Longleftrightarrow \left(\sum_{k=2^{n-1}+1}^{2^n} |f_k|^2 \right)^{1/2} = \mathcal{O}(2^{-\alpha n}) \tag{1.3}$$

We will see more enlightening results which will characterize approximation rate to a smoothness order of the target functions. Nevertheless, this result can be helpful to explicitly construct functions in $\mathcal{A}^\alpha$. Besides, it shows that the rate of approximation is smaller as the coefficients of the target function decrease fast.

1.1.2 Nonlinear Approximation

In the case of nonlinear approximation we can use *n-term approximation*. Which approximates an element $f \in \mathcal{H}$ with the space Σ_n consisting of all elements $g \in \mathcal{H}$ that can be expressed as the linear combination of at most n elements of the orthonormal basis $(\eta_k)_{k\geq 1}$.

$$\Sigma_n = \left\{ \sum_{k\in\Lambda} c_k \eta_k \mid \Lambda \subset \mathbb{N}^*, \#\Lambda \leq n, c_k \in \mathbb{R} \right\} \tag{1.4}$$

Notice that, in general, a sum of two elements in Σ_n will not be an element of Σ_n. Therefore, we are clearly in a context of nonlinear approximation.

Definition 1.3. Let $f \in \mathcal{H}$ and $n \in \mathbb{N}^*$. We define the *error of n-term approximation* as

$$\sigma_n(f)_{\mathcal{H}} = \inf_{g\in\Sigma_n} ||f - g||_{\mathcal{H}} \tag{1.5}$$

Definition 1.4. Let $\alpha \in \mathbb{R}_+^*$. We define the functional space $\mathcal{A}^\alpha\left((\Sigma_n)_{n\geq 1}\right)$ as

$$\mathcal{A}^\alpha\left((\Sigma_n)_{n\geq 1}\right) = \left\{ f \in \mathcal{H} \;\middle|\; \sup_{n\geq 1} n^\alpha \sigma_n(f)_{\mathcal{H}} < \infty \right\} \tag{1.6}$$

Analogously to linear approximation, the best approximation in the sense of the *error of n-term approximation* can be constructed using the coefficients of the target function f in the orthonormal basis $(\eta_k)_{k\geq 1}$. Indeed, every elements of Σ_n is a linear combination of at most n elements of the basis $(\eta_k)_{k\geq 1}$ which means that the best approximation of f from Σ_n is given by the the *sub*-sum of $f = \sum_{k\geq 1} f_k \eta_k$ by keeping the n biggest terms (in absolute value). Let denote by $\gamma_n(f)$ the n^{th} largest of the numbers $|f_k|$, then we have $\sigma_n(f)_{\mathcal{H}}^2 = \sum_{k>n} \gamma_k(f)^2$. That's why it is also possible to characterize $\mathcal{A}^\alpha\left((\Sigma_n)_{n\geq 1}\right)$ using the coefficients of f in the orthonormal basis $(\eta_k)_{k\geq 1}$.

Proposition 1.2. *Let $f \in \mathcal{H}$ and $\alpha \in \mathbb{R}_+^*$ then*

$$f \in \mathcal{A}^\alpha\left((\Sigma_n)_{n\geq 1}\right) \iff \gamma_n(f) = \mathcal{O}\left(n^{-\alpha-1/2}\right) \tag{1.7}$$

As it has been said in the previous section, we will see more enlightening results about the characterization of the spaces $\mathcal{A}^\alpha$. However, it illustrates well the distinction between linear and nonlinear approximation as the approximation rate (in nonlinear approximation) does not depend on the order of the elements of the basis. Moreover, this result gives a simple method to construct functions in spaces $\mathcal{A}^\alpha$.

1.1.3 Approximants

Results such as Proposition 1.1 and Proposition 1.2 give information about what kind of functions can we find in the spaces $\mathcal{A}^\alpha$. More precisely, it highlights the link between the rate of approximation and the coefficients of the target function f in the orthonormal basis. Thus, the choice of the basis is decisive if the goal is to find the best way to approximate an element of a Hilbert space. In the case of approximating functions $f \in L_p(\Omega)$, many basis can be chosen : polynomial basis, Fourier basis, Haar system, etc...

1.2 Function approximation

We shall discuss here the application of nonlinear approximation in a Hilbert space for functions $f : \Omega \mapsto \mathbb{R}$ with $\Omega = [0,1[$ or $\Omega = [0,1[^d$.

1.2.1 Approximation spaces

We shall consider the following general setting. $(X, ||\ ||_X)$ is a normed space in which approximation takes place. $(X_n)_{n\geq 1}$ are subspaces of X where the approximants will come from and we introduce $E_n(f)_X = \inf_{g\in X_n} ||f-g||_X$ the *approximation error*. We also make the following assumptions

1. $X_0 = \{0\}$
2. $X_n \subset X_{n+1}$
3. $aX_n = X_n$, $a \in \mathbb{R}^*$
4. $\exists c \in \mathbb{N}^*$, $\forall n \in \mathbb{N}$, $X_n + X_n \subset X_{cn}$
5. Each $f \in X$ has a best approximation from X_n
6. $\forall f \in X$, $\lim_{n\to\infty} E_n(f)_X = 0$

In order to deal with the question : which functions are approximated at a given rate ? We shall consider the following spaces which we will try to characterize in terms of smoothness spaces.

Definition 1.5. Let $q > 0$, $\alpha > 0$ and $f \in X$.
We define the functional space $\mathcal{A}_q^\alpha(X)$ as

$$\mathcal{A}_q^\alpha(X,(X_n)) = \left\{ f \in X \ \middle|\ \sum_{n\geq 1} [n^\alpha E_n(f)_X]^q \frac{1}{n} < \infty \right\} \tag{1.8}$$

We define the case $q = \infty$ by

$$\mathcal{A}^\alpha(X,(X_n)) = \left\{ f \in X \ \middle|\ \sup_{n\geq 1} n^\alpha E_n(f)_X < \infty \right\} \tag{1.9}$$

1.2.2 Classical approximation examples

Piecewise constants approximation

- $\Omega = [0,1[$
- $X = L_p(\Omega)$
- $X_n = \left\{ \sum_{I\in\Lambda} c_I \mathbb{1}_I \ \middle|\ \Lambda \in \mathcal{I}_n(\Omega)\,,\ c_I \in \mathbb{R} \right\}$ where $\mathcal{I}_n(\Omega) = \left\{ \Lambda \subset \mathcal{P}(\Omega) \ \middle|\ \#\Lambda \leq n\ ,\ \bigcup_{I\in\Lambda} I = \Omega \right\}$

Definition 1.6. Let $p \in]0,+\infty]$ and $\alpha \in]0,1]$. We define the functional space $\mathrm{Lip}(\alpha, L_p(\Omega))$ as

$$\mathrm{Lip}(\alpha, L_p(\Omega)) = \left\{ f \in L_p(\Omega) \ \middle|\ \sup_{h\in]0,1[} h^{-\alpha} ||f(.+h) - f||_{L_p[0,1-h[} < \infty \right\} \tag{1.10}$$

Intuitively, the smoothness of the functions in $\mathrm{Lip}(\alpha, L_p(\Omega))$ increases with α.

Proposition 1.3. *Let $p \in]0,+\infty]$, $\alpha \in]0,1]$ and $f \in L_p(\Omega)$ then*

$$\mathcal{A}^\alpha\left(L_p(\Omega),(X_n)\right) = Lip\left(\alpha, L_\tau(\Omega)\right) \quad \textit{where } 1/\tau = \alpha + 1/p \tag{1.11}$$

As expected, the smoother the function is, the quicker the *approximation error* decreases.

Free knot piecewise polynomial approximation

- $\Omega = [0,1[$
- $X_{n,r} = \left\{ \sum_{I\in\Lambda} P_I \mathbb{1}_I \;\middle|\; \Lambda \in \mathcal{I}_n(\Omega)\,,\; P_I \in \mathbb{P}_r \right\}$ where $\mathcal{I}_n(\Omega) = \left\{ \Lambda \subset \mathcal{P}(\Omega) \;\middle|\; \#\Lambda \leq n\,,\; \bigcup_{I\in\Lambda} I = \Omega \right\}$

Definition 1.7 (r^{th} difference operator with step h)**.** Let $f : \Omega \mapsto \mathbb{R}$, $r \in \mathbb{N}^*$ and $h \in \mathbb{R}^d$.
We define the r^{th} difference operator with step h as

$$\Delta_h^r(f) = \sum_{k=0}^{r} (-1)^{r-k} \binom{r}{k} f(. + kh) \tag{1.12}$$

For $x \in \Omega$ if $\bigcup_{k=0}^{r} \{x + kh\} \not\subset \Omega$ then we set $\Delta_h^r(f)(x) = 0$

Definition 1.8 (r^{th} order modulus of smoothness of f)**.** Let $f \in L_p(\Omega)$, $p \in]0,\infty]$ and $t \geq 0$.
We define the r^{th} order modulus of smoothness of f in $L_p(\Omega)$ as

$$\omega_r(f,t)_p = \sup_{|h|\leq t} ||\Delta_h^r(f)||_{L_p(\Omega)} \tag{1.13}$$

We use $\Delta_h^r(f)$ to define the smoothness of f measured with the $L_p(\Omega)$-norm. Notice that, we always have $\omega_r(f,t)_p \xrightarrow[t\to 0]{} 0$ and intuitively, the faster the convergence is, the smoother f is. Thus, we can bring together all functions whose *moduli of smoothness* have a common behaviour.

Definition 1.9 (Besov spaces)**.** Let $\alpha > 0$, $p \in]0,\infty]$ and $q \in \mathbb{R}_+^*$.
We define the Besov space $\mathcal{B}_q^\alpha(L_p(\Omega))$ as

$$\mathcal{B}_q^\alpha(L_p(\Omega)) = \left\{ f \in L_p(\Omega) \;\middle|\; \int_0^\infty \left[t^{-\alpha} \omega_r(f,t)_p \right]^q \frac{dt}{t} < \infty \right\} \text{ where } r = \lfloor \alpha \rfloor + 1 \tag{1.14}$$

We define the case $q = \infty$ by

$$\mathcal{B}_\infty^\alpha(L_p(\Omega)) = \left\{ f \in L_p(\Omega) \;\middle|\; \sup_{t>0} t^{-\alpha} \omega_r(f,t)_p < \infty \right\} \text{ where } r = \lfloor \alpha \rfloor + 1 \tag{1.15}$$

Proposition 1.4. *Let $p \in]0,\infty]$, $r \in \mathbb{N}^*$ and $\alpha \in]0,r[$ then*

$$\mathcal{A}_q^\alpha\left(L_p(\Omega), (X_{n,r})\right) = \mathcal{B}_q^\alpha\left(L_q(\Omega)\right) \quad \textit{where } 1/q = \alpha + 1/p \tag{1.16}$$

Proposition 1.5. *Let $p > 0$ and $\mathcal{F}_n$ the set of n-piecewise analytic functions such that $\mathcal{F}_n \subset L_p(\Omega)$*

$$\forall n \in \mathbb{N}^*,\ \forall \alpha > 0,\ \ \mathcal{F}_n \subset \mathcal{A}^\alpha\left(L_p(\Omega), (X_{n,r})\right) \tag{1.17}$$

This result shows that, using piecewise polynomial approximation, the *approximation error* for piecewise analytic functions decreases faster than any polynomial order.

Rational approximation

- $\Omega = [0,1[$
- $X = L_p(\Omega)$
- $X_n = \left\{ \frac{P}{Q} \;\middle|\; P, Q \in \mathcal{P}_n(\Omega) \right\}$

Proposition 1.6. *Let $p \in]0,\infty]$, $r \in \mathbb{N}^*$ and $\alpha \in]0,r[$ then*

$$\mathcal{A}_q^\alpha\left(L_p(\Omega), (X_n)\right) = \mathcal{B}_q^\alpha\left(L_q(\Omega)\right) \quad \textit{where } 1/q = \alpha + 1/p \tag{1.18}$$

Surprisingly, in the sense of approximation classes, with one variable there is no difference between *polynomial approximation* and *rational approximation*.

1.3 Optimal approximation error

We shall discuss fundamental results regarding the limitations of linear and nonlinear approximation as it is presented in the article [2], that it to say, for univariate approximation. Then, we will go a little bit further in order to set a reference point to judge the performance of neural networks in the context of approximation.

1.3.1 Univariate approximation $\Omega = [0, 1[$

Definition 1.10 (Kolmogorov n-width)**.** Let X a Banach space and K a compact subset of X. We define the *Kolmogorov n-width* of K as

$$d_n(K) = \inf_{\dim(X_n)=n} \sup_{f \in K} E(f, X_n)_X \tag{1.19}$$

So $d_n(K)$ measures the performance of the best n-dimensional space on the class K. We can use the *Kolmogorov n-width* to determine the limits of linear methods for approximation.

Definition 1.11. Let u and v be two expressions depending on one variable. We define the following notation

$$u \asymp v \iff \exists\, C_1, C_2 \in \mathbb{R} \;\; C_1 u \leq v \leq C_2 u \tag{1.20}$$

Definition 1.12. Let $p \in]0, +\infty]$, $\alpha \in]0, 1]$ and $q \in \mathbb{R}_+^*$, we shall denote by $U_r^\alpha\left(L_p(\Omega)\right)$ the unit ball of $\mathcal{B}_r^\alpha\left(L_p(\Omega)\right)$

Proposition 1.7. *Let $p \in]0, +\infty]$, $\alpha \in]0, 1]$ and $r \in \mathbb{R}_+^*$*

$$d_n\Big(U_r^\alpha\left(L_p(\Omega)\right)\Big) \asymp n^{-\alpha} \tag{1.21}$$

This result shows that, for these spaces, we cannot expect a better *approximation error* than $\mathcal{O}\left(n^{-\alpha}\right)$. In fact, the classical methods of approximation such as splines or wavelets provide such an order of *approximation error*. We can also extend this analysis in the context of nonlinear approximation to see the potential benefit of these methods.

Definition 1.13 (Manifold n-width)**.** Let X a Banach space and K a compact subset of X. We define the *Manifold n-width* of K as

$$\delta_n(K, X) = \inf_{a, M} \sup_{f \in K} ||f - M(a(f))||_X \tag{1.22}$$

The infimum is taken over all continuous mapping $M : \mathbb{R}^n \mapsto X$ and $a : K \mapsto \mathbb{R}^n$

So $\delta_n(K, X)$ quantifies how well the set K can be approximated by n-dimensional nonlinear manifolds. The restriction that the approximation arises through a continuous parameter selection a is essential. Otherwise, we would always have $\delta_n(K, X) = 0$. Besides, the related approximations would not be reasonable objects in the sense that a small variation of the target function f would lead to a completely different parametrization $a(f)$.

Proposition 1.8. *Let $p \in]0, +\infty]$, $\alpha \in]0, 1]$ and $r \in \mathbb{R}_+^*$*

$$\delta_n\Big(U_r^\alpha\left(L_p(\Omega)\right)\Big) \asymp n^{-\alpha} \tag{1.23}$$

This result shows that we cannot obtain a better order of *approximation error* for these spaces than what we obtain via n-term polynomial approximation. More surprisingly, it seems that nonlinear approximation is not superior to linear approximation. However, in these n-width we are looking for the *approximation error* of the worst target function f.

1.3.2 Multivariate approximation $\Omega = [0,1[^d$

The previous results can be generalized in the case of multivariate approximation. In particular, we shall focus on the nonlinear case.

Proposition 1.9. *Let $p \in]0,+\infty]$, $\alpha \in]0,1]$ and $r \in \mathbb{R}_+^*$*

$$\delta_n\Big(U_r^\alpha\left(L_p(\Omega)\right)\Big) \asymp n^{-\alpha/d} \tag{1.24}$$

Notice that the n-width is larger as d increases. This illustrates a phenomenon called the *curse of dimentionality*: the more variables f depends on, the more difficult it is to approximate f.

This result gives an estimation of the optimal *approximation error* we can expect with a given n (which generally corresponds to a measure of complexity of the approximants). However, it can be enlightening to see this from a different point of view: What n do we need to obtain an optimal *approximation error* lesser than a given ε ?

Proposition 1.10. *Let $p \in]0,+\infty]$, $\alpha \in]0,1]$ and $r \in \mathbb{R}_+^*$*

$$\exists C \in \mathbb{R},\ \ \delta_n\Big(U_r^\alpha\left(L_p(\Omega)\right)\Big) \leq \varepsilon \Longleftrightarrow n \geq C\varepsilon^{-d/\alpha} \tag{1.25}$$

Chapter 2

Definition of neural networks

Neural networks can easily be described as a tuple [1]:

$$\Phi = ((T_1, \alpha_1), ..., (T_L, \alpha_L)) \tag{2.1}$$

Where $\forall i = 1, ..., L$, $T_i : \mathbb{R}^{N_{i-1}} \to \mathbb{R}^{N_i}$ is an **affine-linear function** and $\alpha_i : \mathbb{R}^{N_i} \to \mathbb{R}^{N_i}$ is (generally) a non-linear function called **activation function** and $\alpha_L = id_{\mathbb{R}^{N_L}}$.
Plus, we say that Φ has L **layers** including $L-1$ **hidden layers**. Those ones are the layers indexed by $i = 1, ..., L-1$.

Furthermore, we define

- The **input-dimension** $d(\Phi) = N_0$ and the **output-dimension** $k(\Phi) = N_L$.
- The **depth** $L(\Phi) = L$, i.e the number of layers.
- The **number of hidden neurons** $N(\Phi) = \sum_{i=1}^{L-1} N_i$.
- The **number of connections/weights** $W(\Phi) = \sum_{i=1}^{L} ||T_i||_0$, thus if $T_i = A_i x + b_i$ with a matrix A and a (bias) vector b, then $||T_i||_0 = ||A_i||_0 = \sum_{i,j} \mathbf{1}_{A_{ij} \neq 0}$. For generalising to the number of components, we define $W_0(\Phi) = \sum_{i=1}^{L} (||T_i||_0 + ||b_i||_0)$

This network has an input $x \in \mathbb{R}^{N_0}$ and an output $y \in \mathbb{R}^{N_L}$. Between those, the input is evaluated step by step by affine-linear function T_i then by the activation function α_i.
This process can be summarised by the equation:

$$y = R(\Phi)(x) \tag{2.2}$$

Where $R(\Phi) := \alpha_L \circ T_L \circ ... \circ \alpha_1 \circ T_1$ is the **realisation** of the neural network Φ.

2.1 Properties of neural networks

Let Φ be a neural network, does a "shorter" neural network which has the same realisation function of Φ exist? The proposition below ensures this existence [1]:

Proposition 2.1. *There exists a neural network $\hat{\Phi}$ satisfying:*

$$\begin{cases} R(\hat{\Phi}) = R(\Phi) \\ L(\hat{\Phi}) \leq L(\Phi) \\ N(\hat{\Phi}) \leq N(\Phi) \\ W(\hat{\Phi}) \leq M(\hat{\Phi}) \leq k(\Phi) + 2W(\Phi) \end{cases}$$

Remark - it can be $\hat{\Phi} = \Phi$
To make sure that the realisation of Φ is not constant, we can compare the number of weights with length.

Proposition 2.2. *If $W(\Phi) < L(\Phi)$, then $R(\Phi)$ is constant.*

Remark - When the inequality above is satisfied, there exists a layer which has no connection with the next layer. By composition, the realisation is constant, i.e $\forall x, y, R(\Phi)(x) = R(\Phi)(y)$.

We are now focusing on one particular type of neural networks which handles one type of activation function:

Definition 2.1. Let $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, Φ is a **ϱ-network** if:
$\forall i \in [|1, L-1|], \alpha_i = (\varrho_1^i, ..., \varrho_{N_i}^i)$ where $\forall j \in [|1, N_i|], \varrho_j^i \in \{id_{\mathbb{R}}, \varrho\}$.
Φ is a **strict ϱ-network** if $\forall (i,j), \varrho_j^i = \varrho$.

2.2 Approximate functions using Deep Learning

2.2.1 Principle

Deep learning is a field in which scientists use neural networks. It consists in approximating the function $f : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ with a neural network Φ that satisfies $R(\Phi) \simeq f$.

Rigorously, by considering a **loss function** $\mathcal{L} : \mathbb{R}^{N_L} \times \mathbb{R}^{N_L} \to \mathbb{R}$, a penalisation $\mathcal{P}$ and a **learning set** $(x_j, f(x_j))_{1 \leq j \leq n}$, we're seeking out the following minimum [1]:

$$min_\Phi \sum_{i=1}^{n} \mathcal{L}(R(\Phi)(x_i), f(x_i)) + \lambda \mathcal{P}(\Phi) \tag{2.3}$$

The neural network Φ^* that satisfies (2.3) is the best approximation of f on the learning set using Deep learning with the regularisation's constraint $\mathcal{P}$.

2.2.2 Approximation tools

Typical case of function space is **quasi-Banach** set X, supplied by a **quasi-norm** $||.||_X$

example - Let $X = L_p(\Omega)$ be a quasi-Banach set with $p \in]0, +\infty[$ and $\Omega \subset \mathbb{R}^{N_0}$ supplied by the quasi-norm $||\mathrm{f}||_p = (\int_\Omega |f(x)|^p dx)^{\frac{1}{p}}$.

Definition 2.2. $(X, ||.||_X)$ a quasi-Banach space, let Γ a non-empty subset of X and $f \in X$. Then, the **error of best approximation** of f is defined by:

$$E_\Gamma(f)_X = inf_{g \in \Gamma} ||f - g||_X$$

Now, if we have a family of arbitrary subsets $\Sigma = (\Sigma_n) \in \mathcal{P}(X)^{\mathbb{N}}$ and $\alpha \in]0, +\infty[$, if Σ satisfies ideal conditions, we consider the quasi-norm:

$$||f||_{A_q^\alpha(X,\Sigma)} := \begin{cases} (\sum_{i=0}^{\infty} [n^\alpha E(f, \Sigma_{n-1})_X]^q \frac{1}{n})^{\frac{1}{q}}, & 0 < q < +\infty \\ sup_{n \in \mathbb{N}^*} n^\alpha E(f, \Sigma_{n-1})_X, & q = +\infty \end{cases}$$

With that definition, we consider the following subset of X :

$$A_q^\alpha(X, \Sigma) := \{f \in X : ||f||_{A_q^\alpha(X,\Sigma)} < +\infty\}$$

Remark - We admit that if some properties on Σ are satisfied, including sequence (Σ_n) is growing, $(A_q^\alpha(X, \Sigma), ||.||_{A_q^\alpha(X,\Sigma)})$ can be considered as a quasi-Banach space, and also there exists $C > 0$ such as $||.||_{A_q^\alpha(X,\Sigma)} \leq C||.||_X$ (continuous embedding between $A_q^\alpha(X, \Sigma)$ and X).

2.3 Approximation spaces for Deep Learning

2.3.1 Definitions and restrictions

Definition 2.3. Let ϱ be an activation function, $L : \mathbb{N}^* \to \mathbb{N}^* \cup +\infty$ a non decreasing function called **growth**.
We define $NN_{W,L,N}^{\rho,d,k} = \{R(\Phi)/\Phi$ is a ϱ-network with d inputs and k outputs with number of weights, depth and number of hidden neurons respectively inferior to $W, L, N\}$.
Then we define the **approximation families**:

$$W_n(X, \varrho, L) := X \bigcap NN_{n,L(n),\infty}^{\varrho,d,k} \tag{2.4}$$

$$N_n(X, \varrho, L) := X \bigcap NN_{\infty,L(n),n}^{\varrho,d,k} \tag{2.5}$$

These are spaces used for approximating a function $f \in X$ by the realisation of a neural network. Now, let's define the **approximation spaces** :

$$W_q^\alpha := A_q^\alpha(X, \Sigma), \Sigma_n = W_n(X, \varrho, L) \tag{2.6}$$

$$N_q^\alpha := A_q^\alpha(X, \Sigma), \Sigma_n = N_n(X, \varrho, L) \tag{2.7}$$

Remark - When one considers only strict ϱ-network, one adds an S on each sets, for example SN_q^α.

Proposition 2.3. $(W_q^\alpha, ||.||_{W_q^\alpha})$ *(resp* $(SW_q^\alpha, ||.||_{SW_q^\alpha})$*) and* $(N_q^\alpha, ||.||_{N_q^\alpha})$ *(resp* $(SN_q^\alpha, ||.||_{SN_q^\alpha})$*) are quasi-Banach spaces.*

2.3.2 Convergence between generalized and strict ϱ-networks

Is there a difference between ϱ and strict networks? The following theorem states there is no real difference if X satisfies some conditions.

Proposition 2.4. *X is a quasi-Banach set of functions $\Omega \subset \mathbb{R}^d \to \mathbb{R}^k$, if:*

- *Ω is bounded*
- *ϱ can represent the identity with $m \in \mathbb{N}^*$ terms.*

Then, $||.||_{SW_q^\alpha}$ and $||.||_{W_q^\alpha}$ (resp $||.||_{SN_q^\alpha}$ and $||.||_{N_q^\alpha}$) are equivalent.
Furthermore, $SW_q^\alpha = W_q^\alpha$ and $SN_q^\alpha = N_q^\alpha$.

2.3.3 Towards growth functions

Definition 2.4. Let $\mathcal{L}$ and $\mathcal{L}'$ be growth functions:

- **$\mathcal{L}$ is dominated by $\mathcal{L}'$**, i.e. $\mathcal{L} \preceq \mathcal{L}'$, if there are $c, n_0 \in \mathbb{N}^*$ that satisfy $\forall n \geq n_0, \mathcal{L}(n) \leq \mathcal{L}'(c.n)$.
- **$\mathcal{L}$ and $\mathcal{L}'$ are equivalent**, i.e. $\mathcal{L} \sim \mathcal{L}'$, if $\mathcal{L} \preceq \mathcal{L}'$ and $\mathcal{L}' \preceq \mathcal{L}$.

The equivalence of depth functions implies the equivalence of approximation spaces:

Proposition 2.5. *With previous notations, $\alpha > 0$, $q \in]0; +\infty]$, if $\mathcal{L} \sim \mathcal{L}'$, then:*
$||.||_{W_q^\alpha(X,\varrho,\mathcal{L})}$ *and* $||.||_{W_q^\alpha(X,\varrho,\mathcal{L}')}$ *are equivalent, same result for* $||.||_{N_q^\alpha(X,\varrho,\mathcal{L})}$ *and* $||.||_{N_q^\alpha(X,\varrho,\mathcal{L}')}$*. Furthermore:*

$$W_q^\alpha(X, \varrho, \mathcal{L}) = W_q^\alpha(X, \varrho, \mathcal{L}')$$

$$N_q^\alpha(X, \varrho, \mathcal{L}) = N_q^\alpha(X, \varrho, \mathcal{L}')$$

The same result holds for strict ϱ-networks.

2.3.4 The activation function

This paragraph deals with the activation function $\varrho : \mathbb{R} \to \mathbb{R}$. *How do we choose ϱ for correctly approximate a function ?*
Good results are shown for activation functions that are in fact non-linear.

Definition 2.5. ϱ is a *non-degenerate* if the following holds:

1. $\varrho : (\mathbb{R}, Bor(\mathbb{R})) \to (\mathbb{R}, Bor(\mathbb{R}))$ is measurable.
2. ϱ is locally bounded, i.e. bounded on $[-R; R], \forall R > 0$.
3. There is a closed null-set $A \subset \mathbb{R}$ such that ϱ is continuous on $\mathbb{R} \setminus A$.
4. For all polynomial function $p : \mathbb{R} \to \mathbb{R}$, $\varrho \neq p$.

Remark - If ϱ is continuous, then:

$$\varrho \text{ is non-degenerate } \iff \varrho \text{ is non-polynomial.}$$

Example - The **ReLU** functions $\sigma_r : x \to max(0, x^r), r \in \mathbb{N}^*$ are **non-degenerate activation functions** because they are continuous and non-polynomial.

Proposition 2.6. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuous function, $K \subset \mathbb{R}^d$ be a compact, $\varrho : \mathbb{R} \to \mathbb{R}$ be a **non-degenerate** activation function, $\epsilon > 0$, $X = L_\infty(K)$. Then, there exists $g : x \to \sum_{j=1}^{N} c_j.\varrho(< w_j, x > +b_j)$ with appropriate N and (c_j, w_j, b_j) such that:*

$$||f - g||_{L_\infty(K)} \leq \epsilon \tag{2.8}$$

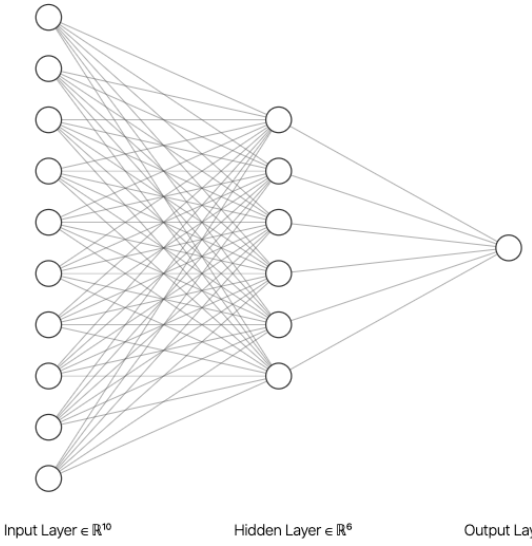

Table 2.1 – g's neural network architecture for $d = 10, N = 6$

Let's generalize this result:

Proposition 2.7. *(Density)*
*Let $\mathcal{L}$ be a growth function and $L = sup_{n \in \mathbb{N}^*} \mathcal{L}(n) \cup \{+\infty\}$, let's assume that $L \geq 2$ and ϱ is a **non-degenerate** activation function, $X = L_p(\Omega) := \{f : \Omega \subset \mathbb{R}^d \to \mathbb{R}^k : ||f||_p < +\infty\}$ with ω bounded.*

1. *For $0 < p < +\infty$, $\Sigma_\infty(L_p(\Omega), \varrho, \mathcal{L})$ is dense in $L_p(\Omega)$.*
2. *For $p = +\infty$, $\Sigma_\infty(L_p(\Omega), \varrho, \mathcal{L})$ is dense in $L_p(\Omega)$ if ϱ is continuous.*

Chapter 3

Error bounds for approximations with deep ReLU networks

The article [**3**] studies the expressive power of shallow and deep neural networks with a piecewise linear activation function. It establishes upper and lower bounds for the network complexity in the setting of approximation in Sobolev spaces.

3.1 Preliminary definitions and settings

3.1.1 Expressive power

According to article [**5**], part of the problem of neural network expressivity consists in characterizing how structural properties of a neural network family affect the functions it can compute.

3.1.2 The ReLU network model

One of the most used activation function is the **ReLU (Rectified Linear Unit) activation function** which is: $\sigma_1(x) = \max(0, x)$

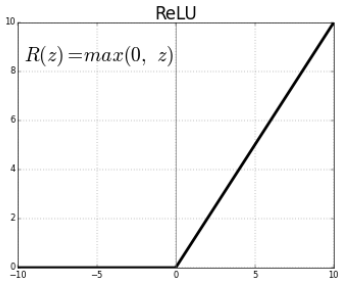

Table 3.1 – ReLU graphical representation

Paper [**3**] considers a feedforward neural network with the ReLU activation function. Indeed, this paper proves that using the ReLU activation function is not much different from using any other **piece-wise linear activation function with finitely many breakpoints**.

Here below is an example of a feedforward neural network representation illustrate the kind of neural network architectures which are studied in paper [**3**].

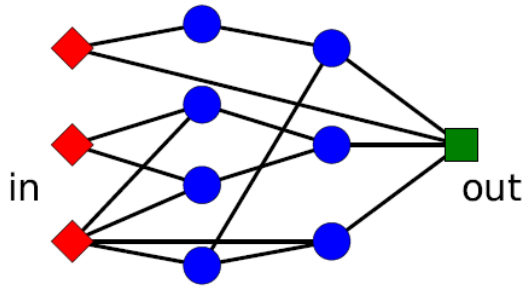

Table 3.2 – A feedforward neural network example

The network consists of several input units, one output unit, and several hidden computation units. Each hidden unit computes:

$$y = \sigma_1(\sum_{k=1}^{N} w_k x_k + b)$$

w_k: weight (adjustable parameter)
b, which depends on the units
x_k: inputs of a computation unit in a certain layer. Here, connections between units in non-neighboring units are allowed.

3.1.3 Approximation error

Paper **[3]** studies the approximation of functions $f : [0,1]^d \to \mathbb{R}$ by ReLU networks. Given a function $f : [0,1]^d \to \mathbb{R}$ and its approximation $\widetilde{f}$, by the approximation error is the uniform maximum error:

$$\|f - \widetilde{f}\|_\infty = \max_{\mathbf{x}\in[0,1]^d} |f(\mathbf{x}) - \widetilde{f}(\mathbf{x})|$$

3.1.4 Sobolev spaces

For the two last two chapters, functions we want to approximate will be on Sobolev Spaces (which are quasi-Banach spaces), e.g. Sobolev spaces in $[0,1]^d$ are:

$$W^{\alpha,\infty}([0,1]^d) = \{f \in L^\infty([0,1]^d) : \forall |k| \leq \alpha \; D^k f \in L^\infty([0,1]^d)\}$$

They also are:

$$W^{\alpha,\infty}([0,1]^d) = \{f \in C^{\alpha-1}([0,1]^d) : \; D^{\alpha-1} f \text{ Lipschitz-continuous}\}$$

The norm in $\mathcal{W}^{\alpha,\infty}\left([0,1]^d\right)$ can be defined by:

$$\|f\|_{\mathcal{W}^{\alpha,\infty}([0,1]^d)} = \max_{\mathrm{A}:|\mathrm{A}|\leq\alpha} \operatorname*{ess\,sup}_{\mathbf{x}\in[0,1]^d} \left|D^{\mathrm{A}} f(\mathbf{x})\right|$$

where $\mathrm{A} = (\alpha_1, \ldots, \alpha_d) \in \{0, 1, \ldots\}^d, |\mathrm{A}| = \alpha_1 + \ldots + \alpha_d$, and $D^n f$ is the respective weak derivative. The unit ball in $\mathcal{W}^{\alpha,\infty}\left([0,1]^d\right)$ is:

$$F_{\alpha,d} = \left\{f \in \mathcal{W}^{\alpha,\infty}\left([0,1]^d\right) : \|f\|_{\mathcal{W}^{\alpha,\infty}([0,1]^d)} \leq 1\right\}$$

3.2 Deep vs. shallow ReLU approximations of smooth functions

The results from article **[3]** show that deep ReLU networks more efficiently express smooth functions than shallow ReLU networks. Indeed, here below are some approximations results showing this assertion.

Functions from the Sobolev space $W^{\alpha,\infty}([0;1]^d)$ can be ϵ-approximated by ReLU networks with depth $\mathrm{O}(\ln(1/\epsilon))$ and the number of computation units $O(\epsilon^{-d/\alpha}\ln(1/\epsilon))$.

On the contrary, a nonlinear function from $C^2([0,1]^d)$ cannot be ϵ-approximated by a ReLU network of fixed depth L with the number of units less than $c\epsilon^{-1/(2L-4)}$.

Thus, in terms of the required number of computation units, unbounded-depth approximations of functions from $W^{\alpha,\infty}([0;1]^d)$ are asymptotically strictly more efficient than approximations with a fixed depth L at least when $\frac{d}{\alpha} < \frac{1}{2L-4}$ and $\alpha > 2$.

The efficiency of depth is even more pronounced for very smooth functions such as polynomials (for example $f(x) = x^2$), which can be implemented by deep networks using only $O(\ln(1/\epsilon))$ units.

3.3 Continuous model selection vs. function-dependent network architectures

When approximating a function by a neural network, there are three possible network architectures: fixed architecture with continuous selection of weights, fixed architecture with unconstrained selection of weights, or adaptive architecture. To a certain extent, the more freedom in the choice of the approximation one has, the higher the expressiveness is.

For example, the complexity of ϵ-approximation of functions from the unit ball $F_{1,1}$ in $W^{1,\infty}([0;1])$ is lower bounded by $\frac{c}{\epsilon}$ in the scenario with a fixed architecture and continuously selected weights. On the other hand, we show that this complexity is upper bounded by $O(\frac{1}{\epsilon\ln(1/\epsilon)})$ if we are allowed to adjust the network architecture. This bound is achieved by finite-depth (depth-6) ReLU networks using the idea of reused subnetworks familiar from the theory of Boolean circuits Shannon.

In the case of fixed architecture, article **[3]** has not established any evidence of complexity improvement for unconstrained weight selection compared to continuous weight selection. However, it remarks that already for approximations with depth-3 networks, the optimal weights are known to discontinuously depend, in general, on the approximated function. On the other hand, the article shows that if the network depth scales as $O(\ln^p(1/\epsilon))$, discontinuous weight selection cannot improve the continuous case complexity more than by a factor being some power of $O(\ln(1/\epsilon))$.

3.4 Upper vs. lower complexity bounds

For fixed architectures with continuous selection, $c\epsilon^{-d/\alpha}$ is a lower bound, and $O(\epsilon^{-d/\alpha}\ln(1/\epsilon))$, an upper bound, so these bounds are tight up to a factor $O(\ln(1/\epsilon))$. Thus, neural networks are able to have upper and lower bounds close to the optimal bound which is $O(\epsilon^{-d/\alpha})$ according to Chapter 1, ignoring a $\ln(1/\epsilon))$ factor.

In the case of fixed architecture with unconstrained selection, under assumption that the depth is constrained by $O(\ln^p(1/\epsilon))$ with any $p \geq 0$, a lower bound is $O(c\epsilon^{-d/\alpha}\ln^{-(2p+1)}(1/\epsilon))$. Without this depth constraint, we only have the significantly weaker bound $c\epsilon^{-d/2\alpha}$.

In the case of adaptive architectures, the upper bound $O(\frac{1}{\epsilon\ln(1/\epsilon)})$ is given for d = n = 1. The lower bound, proved for general d, α, only states that there are $f \in W^{\alpha,\infty}([0;1]^d)$ for which the complexity is not $O(\epsilon^{-d/9\alpha})$.

3.5 ReLU vs. smooth activation functions

A popular general-purpose method of approximation is shallow (depth-3) networks with smooth activation functions (e.g., logistic sigmoid). Upper and lower approximation complexity bounds for these networks show that complexity scales as $\sim \epsilon^{-d/\alpha}$ up to some $\ln(1/\epsilon)$ factors. Comparing this with our bounds, it appears that deep ReLU networks are roughly (up to $\ln(1/\epsilon)$ factors) as expressive as shallow networks with smooth activation functions.

3.6 Conclusion

Here above are several upper and lower bounds for the expressive power of deep ReLU networks in the context of approximation in Sobolev spaces. However, this setting may not quite reflect typical real-world applications, which usually possess symmetries and hierarchical and other structural properties substantially narrowing the actually interesting classes of approximated functions.

Chapter 4

Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms

In this chapter, we generalize the above results with fonction $\in R^{k^{R^d}}$ with $k \in \mathbb{N}$

4.1 General feedforward architecture

We can define a more general architecture for feedforward neural networks that also allows for connections of neurons in non-neighboring layers. Because the transformation to have the layers l is a linear function or the above layers, we can represent the transformation with a matrix A_l and a vector b_l depending on $l \in 1...L$ where L is the number of layers. A neural network with input dimension d and L layers is then a sequence of matrix-vector tuples :

$$\Phi = ((A_1, b_1), (A_2, b_2), \cdots, (A_L, b_L))$$

Secondly we have also an activation function, but in our case, it will be always the same for each layer : the ReLU function, which acts componentwise:

$$\varrho : x \longrightarrow \max\{0, x\}$$

If N_l is the number of neurons in the layer l (then $N_0 = d$), we can decompose A_l matrix like this :

$$A_l = \left(A_{l,x_0} | A_{l,x_1} | \cdots | A_{l,x_{l-1}}\right) \in N_l \times \sum_{i=0}^{l-1} N_i$$

Then we have the process for compute $R_\varrho(\Phi)(x) = x_L$:

$$\begin{aligned} &x_0 := x, \\ &x_l := \varrho\left(A_{l,x_0} x_0 + \cdots + A_{l,x_{l-1}} x_{l-1} + b_l\right), \qquad \text{for } l = 1, ..., L-1 \\ &x_L := A_{L,x_0} x_0 + \cdots + A_{L,x_{L-1}} x_{L-1} + b_L \end{aligned}$$

As same as chapter 1, we define

- The **Number of layers** $L(\Phi) = L$
- The **number of neurons** $N_{tot}(\Phi) = d + \sum_{i=1}^{L-1} N_i = d + N(\Phi)$.
- The **number of weights** $M(\Phi) = \sum_{i=1}^{L} (||A_i||_{l^0} + ||b_i||_{l^0})$

The most important number is $M(\Phi)$, because it is the number of parameters we will adjust during the learning process. We aim at reducing this number.
We define also the architecture $\mathcal{A}$ of a neural network Φ which is basically the same neural network but where non-zero weights of Φ is a 1 in the architecture $\mathcal{A}$.
We can see that this approach is a generalisation of chapter 1, because if we choose:

$$A_l = \left(0_{N_l \times \sum_{i=0}^{l-2} N_i} | A_{l,x_{l-1}}\right), \text{ for } l = 1, ..., L$$

we recover the same pattern. This type of neural network is called standard neural network and it is proved that for any neural network Φ, we can find a Φ_{std} which is a standard one, for which $R_\varrho(\Phi)(x) = R_\varrho(\Phi_{std})(x)$ for any $x \in \mathbb{R}^d$.

4.2 Neural Network operations

We can define :

- Concatenation : Let Φ^1 and Φ^2, then $\Phi^1 \bullet \Phi^2$ is defined so that $R_\varrho(\Phi^1 \bullet \Phi^2)(x) = R_\varrho(\Phi^1) \circ R_\varrho(\Phi^2)(x)$ for all $x \in \mathbb{R}^d$.
- Identity : We can find a Φ^{Id} so that $R_\varrho(\Phi^{Id}) = Id_{\mathbb{R}^d}$.
- Sparse concatenation : Let Φ^1 and Φ^2, then $\Phi^1 \odot \Phi^2 = \Phi^1 \bullet \Phi^{Id} \bullet \Phi^2$
- Parallelization : Let Φ^1 and Φ^2, then $P(\Phi^1, \Phi^2)$ is the parallelization neural network from Φ^1 and Φ^2. In fact, it is just grouping of both network, so they take the same entry and the result is the grouping of both result.

These operations are very useful to prove the following results. We won't demonstrate them in this paper.

4.3 Sobolev spaces

Definition Sobolev spaces : Let $\alpha \in \mathbb{N}$ and $1 \leq p \leq \infty$. Then :

$$W^{\alpha,p}(\Omega) := \left\{ f \in L^p(\Omega) : D^k f \in L^p(\Omega) \text{ for all } k \in \mathbb{N}^d \text{ with } |k| \leq \alpha \right\}$$

equipped with the norm:

$$||f||_{W^{\alpha,p}(\Omega)} := \left(\sum_{0 \leq k \leq \alpha} ||D^k f||^p_{L^p(\Omega)} \right)^{1/p}$$

and

$$||f||_{W^{\alpha,\infty}(\Omega)} := \max_{0 \leq k \leq \alpha} ||D^k f||_{L^\infty(\Omega)}$$

Definition Sobolev-Slobodeckij spaces : $0 < s < 1$ and $1 \leq \infty$

$$W^{s,p}(\Omega) := \left\{ f \in L^p(\Omega) : ||f||_{W^{s,p}(\Omega)} < \infty \right\}$$

with

$$||f||_{W^{\alpha,p}(\Omega)} := \left(||f||^p_{L^p(\Omega)} + \int_\Omega \int_\Omega \left(\frac{|f(x) - f(y)|}{|x - y|^{\alpha + d/p}} \right)^p dxdy \right)^{1/p}$$

and

$$||f||_{W^{\alpha,p}(\Omega)} := \max \left\{ ||f||^p_{L^\infty(\Omega)}, \text{ess} \sup_{x,y \in \Omega} \frac{|f(x) - f(y)|}{|x - y|^\alpha} \right\}$$

They are both Banach spaces.

4.4 Results

$$\mathcal{F}_{\alpha,d,p,B} := \left\{ f \in W^{\alpha,p}((0,1)^d) : ||f||_{W^{\alpha,p}((0,1)^d)} \leq B \right\}$$

Theorem Upper complexity bounds: Let $d \in \mathbb{N}, \alpha \geq 2, 1 \leq p \leq \infty, B > 0$ and $0 \leq s \leq 1$ Then there exists $c > 0$ such that for all $\epsilon \in (0, 1/2)$, there is a neural network architecture $\mathbb{A}_\epsilon = \mathbb{A}_\epsilon(d, \alpha, p, B, s, \epsilon)$ with d inputs and on-dimensional output such that for any $f \in \mathcal{F}_{\alpha,d,p,B}$ there is a neural network Φ_ϵ^f that has architecture $\mathbb{A}_\epsilon$ such that

$$||R_\varrho(\Phi_\epsilon^f) - f||_{W^{s,p}((0,1)^d)} \leq \varepsilon$$

and

(i) $L(\mathcal{A}_\epsilon) \leq c \log_2(\epsilon^{-\alpha/(\alpha-s)})$

(ii) $M(\mathcal{A}_\epsilon) \leq c\epsilon^{-d/(\alpha-s)} \log_2(\epsilon^{-\alpha/(\alpha-s)})$

(iii) $N(\mathcal{A}_\epsilon) \leq c\epsilon^{-d/(\alpha-s)} \log_2(\epsilon^{-\alpha/(\alpha-s)})$

Theorem Lower complexity bounds : Let $d \in \mathbb{N}, \alpha \geq 2$, $B > 0$ and $k \in \{0, 1\}$ Then there exists $c > 0$ such that if $\epsilon \in (0, 1/2)$ and a neural network architecture $\mathbb{A}_\epsilon = \mathbb{A}_\epsilon(d, \alpha, p, B, s, \epsilon)$ with d inputs and one-dimensional output such that for any $f \in \mathcal{F}_{\alpha,d,\infty,B}$ there is a neural network Φ_ϵ^f that has architecture $\mathbb{A}_\epsilon$ such that

$$||R_\varrho(\Phi_\epsilon^f) - f||_{W^{k,\infty}((0,1)^d)} \leq \varepsilon$$

then

$$M(\mathcal{A}_\epsilon) \geq c\epsilon^{-d/2(\alpha-k)}$$

4.5 Conclusion

$$M(\Phi) = \mathcal{O}(\epsilon^{-d/(\alpha-s)})$$

- These bounds give us a bracket of complexity for neural network architectures used in Sobolev training.
- Curse of the dimension: It depends strongly on d

List of Tables

Bibliography

Publications

[1] GRIBONVAL R., KUTYNIOK G., NIELSEN M. and VOIGTLAENDER F., *Approximation spaces of deep neural networks*, 2019

[2] DEVORE R., *Nonlinear approximation*, 1998

[3] YAROTSKY D., *Error bounds for approximations with deep ReLU networks*, 2017

[4] GUHRING I., KUTYNIOK G., PETERSON P., *Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms*, 2019

[5] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl Dickstein, *On the Expressive Power of Deep Neural Networks*, 2017