MASTERARBEIT / MASTER’S THESIS

Titel der Masterarbeit / Title of the Master‘s Thesis

„Estimation of sets with Barron boundaries under margin conditions“

verfasst von / submitted by

Simon Fourmann

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2023 / Vienna 2023

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 645
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Master Data Science
Betreut von / Supervisor:	Ass.-Prof. Dr. Philipp Christian Petersen, M.Sc.

Zusammenfassung

Beim maschinellen Lernen geht es häufig um die Konstruktion von Schranken für Probleme, bei denen das Ziel darin besteht, einen Klassifikator mit Funktionen einer bestimmten Klasse aus Stichprobenpunkten zu approximieren. Wie eng die Schranken sind, hängt von den Annahmen über den zu lernenden Klassifikator, die Verteilung der Daten und die Art der als Hypothesen verwendeten Funktionen ab. Ziel dieser Arbeit ist es, solche Schranken für Fälle abzuleiten, in denen der Klassifikator eine Menge mit Funktionen der Barron-Klasse als lokale Grenzen und ein neuronales Netz als Hypothesenmenge ist. Eine weitere und zentrale Annahme wird über die Verteilung der Daten gemacht, nämlich dass eine Randbedingung für die Verteilung gilt. Dies bedeutet, dass in einem Bereich um die Grenze der zu lernenden Menge keine Daten gezogen werden können. Unter diesen Annahmen zeigen wir, dass die obere Schranke für das Risiko optimal ist und nur polynomiell von der Dimension abhängt.

Abstract

Machine learning often deals with the construction of error bounds for problems where the goal is to approximate a classifier with some functions of a certain class from sampled points. The tightness of the bounds depends on the assumptions made about the classifier to learn, the distribution of the data and the type of functions taken as hypotheses. This goal of this paper is to derive such bounds for cases where the classifier is a set with functions of the Barron class as local boundaries and neural network as hypothesis set. A further and central assumption is made about the distribution of the data, namely that a margin condition on the distribution holds. This means that no data can be sampled in an area around the boundary of the set to learn. Under these assumptions we show the upper bound for the risk is optimal and only polynomially dependent of the dimension.

Résumé

L'apprentissage automatique consiste souvent en la construction de borne d'erreur pour des problèmes visant à approximer un classifieur par des fonctions d'un certain type à partir de points échantillonés. L'acuité des bornes dépend des hypothèses faites sur le classifieur à estimer, la distribution suivie par les échantillons et enfin les fonctions servant à l'estimation. Le but de cet article est de dériver ce genre de bornes lorsque le classifieur est un ensemble dont les frontières sont localement des fonctions de la classe Barron et les fonctions utilisées pour l'estimation des réseaux de neurones. Une hypothèse cruciale est également faite, à savoir qu'une condition de marge existe sur les données. Cela veut dire que la probabilité d'être tiré pour des points se situant dans une marge autour de la frontière de l'ensemble à estimer est nulle. Sous ces conditions, nous montrons que la borne supérieure du risque est optimale et dépend seulement de façon polynomiale de la dimension de l'ensemble.

Acknowledgements

I am very grateful to Philipp Petersen who gave me the opportunity to have an insight into how mathematical research works, to challenge and expand my knowledge, and to push drastically further my conceptions of what it is to be *rigorous*. My thanks also go to the whole team I had the chance to work with. Finally, my family and my friends who offered me their support throughout the realization of this thesis.

Vienna, January 2023.

Contents

1 Introduction

1.1 Motivation

In this section we explain the general framework in which we work, namely statistical learning, as well as the specific problem that motivates this work.

1.1.1 Statistical Learning Basics

This part is particularly addressed to readers unfamiliar with statistical learning, and we explain in it the core ideas of this field met in this thesis. The principle of statistical or machine learning is to (as the name tells it) learn something from samples, using an algorithm (and the computational power of a computer). What does *learning* mean and what is this *something* which is learnt ? This something is typically a function that associates to a certain data a certain value. Let us call it *f*. The function f returns for some point x of a set X some other point y of a set Y. In the rest of this section, we will assume that $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$:

$$f : X \to Y,\ f(x) = y.$$

We say that f is a binary classifier if Y is a set of two elements (say $\{0, 1\}$). Note that this a formalization of any process like this one : f is the function which tells from the age of a person whether they qualifies to the reduced tickets at Vienna State Opera. The current policy at this Opera House is that people under 27 benefit from the reduced tickets. The function can hence be written so : "If under 27 (excluded), then benefits ; if not, then does not benefit.", or, denoting by $x \in \mathbb{N}$ the age of the person and the set {does not benefit, benefits } by $Y = \{0, 1\}$:

$$f : \mathbb{N} \to \{0, 1\},\ f(x) = \mathbb{1}_{x<27}(x) = \begin{cases} 1 & \text{if } x < 27, \\ 0 & \text{if } x \geq 27. \end{cases}$$

Now suppose that you know what is the policy lead by the Vienna State Opera but do not have access to the limit age, and that you can only ask people at the end of a performance their age and whether or not they benefited from reduced tickets. Finding an unknown function from samples is precisely what statistical learning does. Formally, you have in this case a hypothesis set consisting of all the functions of the form $h(x) = \mathbb{1}_{x<n}(x),\ n \in \mathbb{N}$, which we call $\mathcal{H}$ and samples that are the people you meet at the end of a performance. It is very important to say that even though the distribution $\mathcal{D}$ of their age can be anything, the assumption made in statistical learning is that the samples are independent and identically distributed (i.i.d). This is a crucial and common assumption, which basically says that the age and ticket status of a person you ask in front of the Opera do not depend on the data of the others you meet and that you have on average for every age always the same amount of people of this age every night. So suppose that you ask ten people and get the following sample : $S = \{(65, 0),\ (34, 0),\ (42, 0),\ (13, 1),\ (71, 0),\ (11, 1),\ (73, 0),\ (23, 1),\ (87, 0),\ (84, 0)\}$. From this sample you can form a hypothesis h_S you hope close to f, using for example the smallest age labelled 0 as guessed limit age (or the oldest labelled 1 if you met only people with reduced tickets). For this sample, it would yield $h_S(x) = \mathbb{1}_{x<34}$. This hypothesis is right on this set, and we say that h_S has an *empirical error* or *empirical risk* equal to zero. Denoting it with $\hat{\mathcal{R}}(h_1)$:

$$\hat{\mathcal{R}}(h_1) = \frac{1}{10} \cdot \#\{(x_i, f(x_i)) \in S : h_1(x_i) \neq f(x_i)\} = 0.$$

This is how statistical learning works : trying to produce hypotheses that minimize the empirical risk on a sample. This principle is called *Empirical Risk Minimization* (ERM). However, suppose that ten percent of the Opera visitors are aged between 27 and 33 and that the distribution is uniform. It means that the hypothesis h_S fails on average to classify rightly every tenth visitor. We call this the *true error* or *risk*, denoted $\mathcal{R}$. Here $\mathcal{R}(h_S) = 0.1$. But suppose that ninety percent of the visitors are between 27 and 33, then the true error of h_S would be 0.9, which is really bad ! Should we conclude that a classifier obtained following an ERM rule has no guarantee to be an actual good classifier ? Fortunately, no. If the proportion of people with age between 27 and 33 were of ninety percent, then the above sample S would be pretty unlikely. The probability of running ten times in a row into a person not in this category has indeed a probability of $(1 - 0.9)^{10} = 10^{-10}$! If the sample were composed of hundred people, such a sample (without people aged between 27 and 33) would have a probability of 10^{-100} etc. Yet there exist plenty of other samples that lead to a classifier that has a big true error while being with a zero empirical risk, therefore more computation is needed to infer the actual risk of a h_S with no empirical error (see [18] for a comprehensive account). But you can see here why the bigger the sample is, the unlikelier it becomes for a classifier with no empirical risk to be a bad classifier in general : this would mean that the samples it was trained on were non representative of the true classification, which turns less and less probable if the sample size grows, thanks to the i.i.d. assumption. This example may give to the reader unfamiliar with these topics a rough insight of the *Fundamental Theorem of Statistical Learning* ([18], Theorem 6.8, detailed in Section 3). This theorem says that there are (quite widespread) configurations where when the empirical risk can be null, the true error is bounded by something that decreases along with the increase of the sample size, namely at a rate of order $\frac{\log(n)}{n}$ where n is the sample size. The main purpose of this work is to show that neural networks learning a certain type of classifier (see Definition 3.4) can fulfill the conditions of the *Fundamental Theorem of Statistical Learning* and therefore can achieve true error rates of $\frac{\log(n)}{n}$.

1.1.2 Neural Networks and *Barron* functions : a Framework

[For all the technical details concerning neural networks, see Subsection 1.6.] A neural network is roughly a highly parametrizable function made of some input and output layers, with some layers in between. Each layer consists of *neurons* or *computational units*. These are functions that typically output something between 0 and 1, whether or not the input exceeds a certain threshold (on of the most used one is the ReLU function). The input for a neuron is a weighted sum of the outputs of the precedent layer neurons. Even if able to approximate extremely complex functions, a neural network stays a recursive sum of ReLU activated linear functions, which makes them easily trainable ([12, 14]).

The Barron class was introduced by the eponymous mathematician in [4, 2, 3]. The functions of this class have a bounded first Fourier moment (see Definition 2.1), which allows them to be approximated by neural networks independently of their ambient dimension (see Proposition 3.2). This is a huge advantage these functions have, regarding the pitfall constituted by the *curse of dimensionality* in machine learning. The issue here is that the sample complexity (the minimal number of samples you need to probably achieve a certain risk) uses to grow exponentially with the ambient dimension and the decrease of the risk. This problem occurs also when dealing with neural networks, concerning their size and the magnitude of their parameters. The asset of the Barron class is a double one, as it addresses the two curses : the one related to the complexity of the model (here neural networks) and the one related to the sample complexity.

1.1.3 Margin Condition

The idea of a margin condition on the distribution roots in the developing of the Support Vector Machine (SVM) in the 1990s (cf. [20]), following a original idea by V. Vapnik in 1963 ([24]). SVM rely on the idea that when learning a classifier, the classes can be linearly separated (if projected in another space, or not). Typically, for a binary classifier in dimension two, the two classes are split into two different half-planes and separated by a line. The assumption goes further than the mere linear separability, since a margin between the classes is assumed : the minimal distance from the separating plane is strictly positive for any sample. The framework evolved since, and the common framework for binary classifiers is as follows : labelled points are generated by two random variables $X \in \mathbb{R}^d$ and $Y \in \{-1, 1\}$ following a joint distribution P. This means that unlike the assumptions in the basic SVM model, the position of $x \sim X$ does not fully determine its label, rather a *posterior probability* is introduced $\eta(x) := P(Y = 1|X = x)$. The posterior probability addresses the quantification of the *noise* in models where the classification highly depends on the coordinates of the data in the space, but not in an absolute way. The concept of *decision line* emerges in this framework as the set of points for which the classification is equivalent to coin-tossing, $D = \partial\{x \in \mathbb{R}^d : \eta(x) < 1/2\}$. In [22, 15, 23] among others (see the next subsection for more related works) the quantification of this noise is tackled. Tsybakov in [23] introduced a first condition, often called *margin condition* although in this thesis we will use a second common designation, *noise condition*. This low-noise condition holds if there exist a $q > 0$ (the *noise exponent*) and some constant $C > 0$ such that for all $t > 0$:

$$P_X(|2\eta(x) - 1| \leq t) \leq Ct^q. \tag{1}$$

There are two ways of understanding this, either as a "spatial" condition or as a "probabilistic" one. This means that this condition can be obtained through condition on the sample distribution X (in this case, the condition states that it is unlikely for a point to be sampled in a noisy area) or on the posterior distribution η (in this case, the bigger the exponent, the more deterministic is the label, *i.e.* the less noise there is). Both interpretations are equivalent, but a second kind of margin condition was introduced in [20] to highlight the "spatial" interpretation. Recalling that $D = \partial\{x \in \mathbb{R}^d : \eta(x) < 1/2\}$, and defining the following distance $\Delta(x, D) = \inf_{x' \in D} ||x - x'||_2$ where we use the Euclidean norm, the margin condition holds if there exist a $p > 0$ (the *margin exponent*) and some constant $C > 0$ such that for all $t > 0$:

$$P_X(\Delta(x, D) \leq t) \leq Ct^p. \tag{2}$$

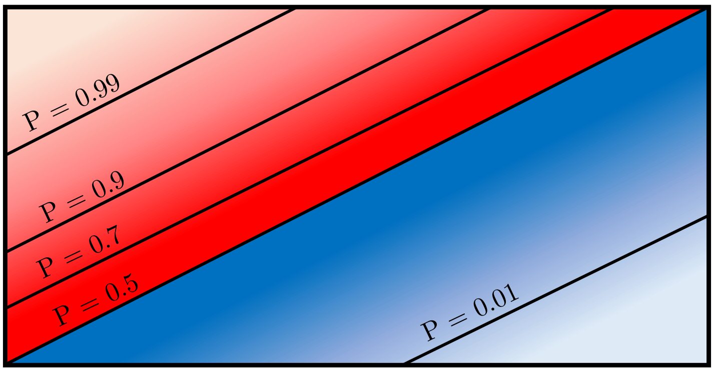

Figure 1: **Noisy case.** The brighter the colors, the closer to 1/2 gets the probability $P := \eta(x)$. Red corresponds to the class 1 and blue to the -1 one.

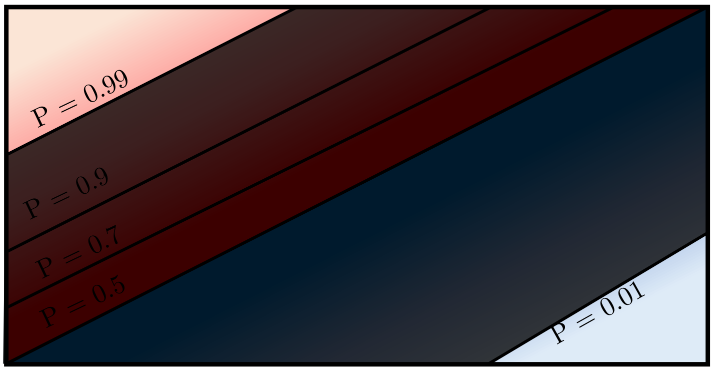

Figure 2: **High noise exponent, "spatial" point of view.** In this figure, the importance of the noise exponent is materialized by the greyed zone on which the probability of being sampled is extremely low.

In this thesis, we are working in a noise-free margined framework, that is with infinite noise and margin exponents. These assumptions ensure the absolutely deterministic characteristic of the classifier to learn and a non zero distance for the samples to the decision line. To fit this framework, we still have to reformulate our model a bit : first, we want to scale the data to have a margin equal to 1 (in order to have a probability of being inside the margin equal to zero) and second to define the following distribution $P(Y = 1|X \in \Omega) = 1$, $P(Y = 1|X \notin \Omega) = 0$ and $P(Y = 1|X \in \partial\Omega) = 1/2$ (note that $P_X(X \in \partial\Omega) = 0$). Thus we can rewrite our margin condition (Definition 2.3) in the terms of [20].

1.2 Related Work

This thesis can be seen as an emphasis on some results of [16] and [9], since it is primarily the evaluation of the results of these two papers with a margin condition on the data sampling. We are therefore following an approach aligned with these papers and the evolving in the same domain. For these reasons we invite the reader to refer to the these papers for a a more detailed overview of the works related to this thesis concerning the neural network part. We still highlight here [4, 2, 3] as these are the papers where the curse of dimensionality breaking capacity of the Barron class was first proven.

Regarding the *margin condition*, [23] by Alexandre Tsybakov is the first paper that formalized the idea of a low-noise condition, which is highly studied, so that it is sometimes referred as the *Tsybakov condition* (see [22, 15, 26, 17, 7, 19, 21, 11, 1]). However, this thesis is focused on the another type of margin condition we introduced (2). This *spatial margin condition* seems to have been first introduced in [20]. It has been since directly studied in [8] where the distinction made between a noise-free environment and the presence of a margin around the boundary for the sampling is crucial. Recently

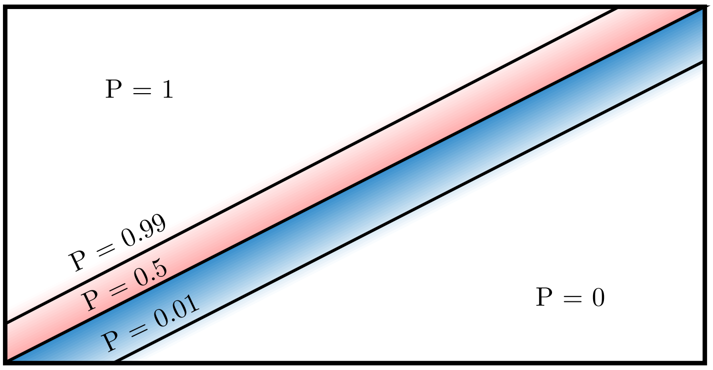

Figure 3: **High noise exponent, "probabilistic" point of view.** Here the high noise exponent pushes the posterior probability η to extreme values (0 or 1), but there is no condition on the location of the sampled points.

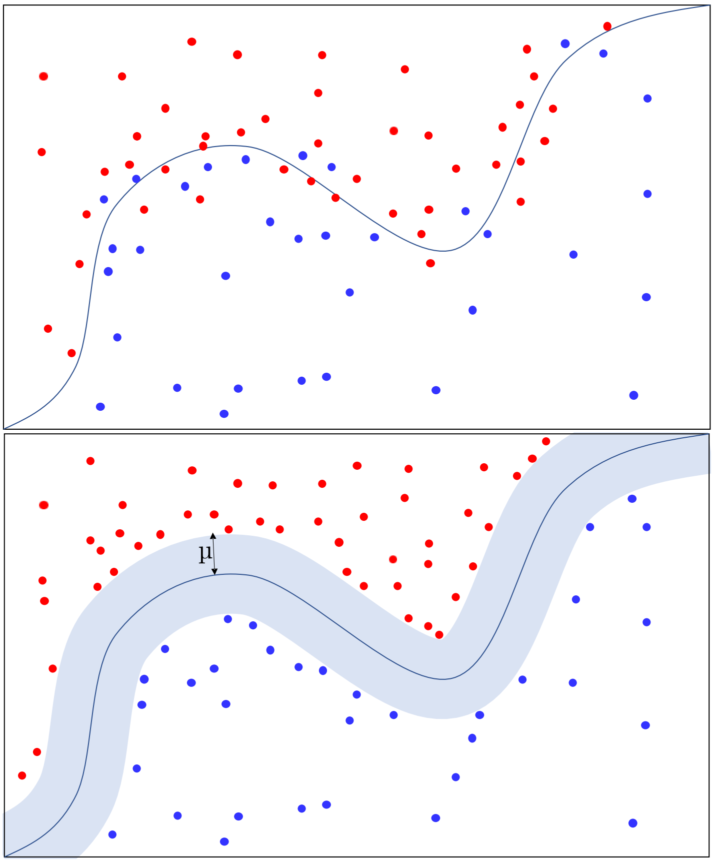

Figure 4: **Sampling in a noisy and unmargined case vs. case with $q = p = \infty$ and $\mu = 1$.** The line represents $\eta(x) = 1/2$ and we see that without condition on the noise, points sampled in a "red" zone are labelled blue and vice-versa. On the contrary, infinite noise and margin exponents ensure no overlapping of the classes and a margin around the boundary.

[13] derived some bounds for the estimation of smooth functions with the help of deep neural networks under the noise and margin conditions. Applied to Lipschitz functions, which is the case of Barron functions (see [10], Theorem 3.3), the bound they derive for the risk is for p and q margin and noise conditions, respectively:

$$\mathcal{R}(h) \lesssim \left(\frac{\log^3 n}{n}\right)^{\frac{q+1}{(q+2)+(d-1)(q+1)/p}}.$$

As written in the above subsection, in this thesis the assumption is that $q = p = \infty$, which would lead to a bound of the order $\frac{\log^3(n)}{n}$.

1.3 Our Contribution

Our major contribution is the Theorem 4.3, *i.e.* the computation of a risk bound of order $\frac{log(n)}{n}$ for a class of deep neural networks learning the characteristic function of a set with piece-wise Barron boundary following an ERM rule with n samples *when a margin condition holds.* This is the best bound possible without more assumptions. This theorem is a direct consequence of the Theorem 3.7 that builds the adequate neural network class, which is a class of neural network whose empirical error is limited around the boundary of the set. Note that neural networks in Theorem 3.7 are a slightly modified version of the one used in [9], Theorem 3.7. The issue with [9] class is that the networks differ from the classifier it learns on a set of bounded measure, that decreases along with the growth of the sample size ; but while having an insignificant risk, there is no guarantee for the distance of the misclassified points from the set boundary, which is problematic to achieve a zero empirical error, even under a margin condition assumption. In total these two theorems yield that in classification problems with margins and Barron functions as boundary, deep neural networks are always optimal without a visible curse of dimension.

1.4 Outline

In Section 2, we begin with a first bound under margin condition in the simple case of a classifier of the form $\mathbb{1}_{f(x_1,\ldots,x_{d-1})>x_d}$ where f is a function of the Barron class. In Section 3, we build a neural network that is a modified version of one obtained in [9], and which approximates well a set with piece-wise Barron boundary, making errors only around the boundary of the said set. In Section 4, we draws the conclusions opened by the network in Section 3, *i.e.*, that ERM in a realizable case with neural network is possible in this situation.

1.5 Discussion

The kind of margin condition we deal with, that is a *margin condition on the distribution* may seem a bit of a cheat-code, since assumptions about the distribution can lead to absolutely everything : the *No-Free-Lunch Theorem* ([18], Theorem 5.1) states roughly that some misleading distribution always exists whereas a distribution giving always the same point ensures the possibility of having a perfect classifier with no risk at all. The question is then whether such an assumption is justified. In the literature, the question, as far as we know, has never been addressed for itself when it comes to neural networks, except in ([25]). However the assumption of a sort of *no-man's-land* between two sets of points with different labels is not new, and has even its dedicated model in supervised learning : the Support Vector Machine (SVM). The assumption of a margin condition on the distribution is just the reverse of a model that assumes the presence of *noise*, and both have their applications in the various fields where machine learning is used.

1.6 Notations

Let us introduce in this section the notations we will use in this thesis :

General notations

- If we denote a number by a Greek letter without giving more precision on the set to which it belongs, this means that it is a real number. For instance "Let $\mu > 0$" implies that $\mu \in \mathbb{R}$.
- On the other hand if we denote a number by a Latin letter without giving more precision on the set to which it belongs, this means that it is an integer. For instance "Let $m > 0$" implies that $m \in \mathbb{N}^*$.

- Any statement beginning with something in the form "Let $a, b, c > 0$" or "Let $f, g \in \mathcal{C}^\infty$" means each time that the condition is valid for *all* the items mentioned. In the examples, the three integers a, b and c are positive and both f and g are smooth functions.
- For $d > 0$, let $x = (x_1, \ \dots \ , x_d) \in \mathbb{R}^d$. For $i \in \mathbb{N}$ we write x_i for the i-th coordinate and x^i designates the vector $x^i = (x_1, \ \dots \ , x_{i-1}, x_{i+1}, \ \dots \ , x_d) \in \mathbb{R}^{d-1}$.
- For $d > 0$, let $a, b \in \mathbb{R}^d$, the notation $[a, b]$ designates the set $[a, b] = \Pi_{i=0}^d [a_i, b_i]$ where we use the Cartesian product.
- For some $n, d > 0$, let $a_n \in \mathbb{R}^d$. Then for $j > 0$, $a_{n,j}$ designates the j-th coordinate of the vector $a_n = (a_{n,1}, \ \dots \ , a_{n,d})$.
- For $m \leq n$, we use the following notation : $[\![m, n]\!] = [m, n] \cap \mathbb{Z} = \{m, m+1, \ \dots \ , n-1, n\}$.

Notations related to functions and sets

- For a set A, $\mathbb{1}_A$ designates the indicator function or characteristic function of the set A :

$$\mathbb{1}_A(x) = \begin{cases} 0 & \text{if } x \notin A, \\ 1 & \text{if } x \in A. \end{cases}$$

- For two sets A and B, the symmetric difference Δ designates the set $A \Delta B = (A \setminus B) \cup (B \setminus A)$.
- For a set A, $\#A$ designates its cardinal.
- For a function $f : X \to Y$, we denote by $||f||_{\text{sup}} := \sup_{x \in X} |f(x)|$.

- For a function $f : X \to Y$, we denote by supp $f :=$

Notations concerning neural networks

- We use σ to denote the ReLU function $\sigma : \mathbb{R} \to [0, \infty)$:

$$\sigma(x) = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{if } x \geq 0. \end{cases}$$

- When mentioning a neural network Φ, we refer to its architecture as follows : $\mathcal{A} = (n_1, ..., n_k)$, where the network Φ has k layers with n_i neurons on each of them. For such an architecture is associated k weights and biases matrices : $(W_i)_{i=1}^k$ and $(b_i)_{i=1}^k$. The weights matrices have the size $n_i \times n_{i-1}$ and biases $n_i \times 1$. Note that the size of the first layer corresponds to the dimension of the input, so $n_1 = d$ if the input is in $\mathbb{R}^d$, and that for a binary classifier, the output layer has only one neuron, so $n_k = 1$. We use the notation $R_\sigma \Phi(x)$ to denote the output (or *realization*) of Φ for $x \in \mathbb{R}^d$ as input. This corresponds to what the following process yields :

$$\begin{aligned} x_0 &= x \in \mathbb{R}^d, \\ x_i &= \sigma \left(\sum_{j=1}^{n_i} W_{i-1} x_{i-1} + b_{i-1} \right), i \in [\![1, k-1]\!], \\ x_k &= \sum_{j=1}^{n_k} W_{k-1} x_{k-1} + b_{k-1}. \end{aligned}$$

2 General Framework for Estimation under Margin Condition

We want to estimate functions of the form $\mathbb{1}_B$ where B is a set with a boundary that belongs to the Barron class (see 2.1). Our goal is to show that if a margin condition holds (*i.e.* no point can belong to an μ-tube around the boundary, see 2.3 below), then for $0 < \alpha \leq \mu$, any subset of the Barron class of functions such that every Barron functions in general is at most α away from a function the set (we call it an α-net) is an adequate hypothesis class, since it contains at least one function that performs exactly the same classification than $\mathbb{1}_B$. In other words, there exists a subset of the class (and we will see hereinafter that it is a substantially smaller one) such that it suffices to look into it to find a classifier doing the same thing as $\mathbb{1}_B$.

First, let us define the *general Barron class* of functions :

Definition 2.1. [[9], Definition 2.1] Let $\emptyset \neq X \subset \mathbb{R}^d$ be bounded. A function $f : X \to \mathbb{R}$ is said to be of *Barron class with constant* $C > 0$, if there are $x_0 \in X$, $c \in [-C, C]$, and a measurable function $F : \mathbb{R}^d \to$ satisfying

$$\int_{\mathbb{R}^d} |\xi|_{X,x_0} \cdot |F(\xi)|\, d\xi \leq C \quad \text{and} \quad f(x) = c + \int_{\mathbb{R}^d} \left(e^{i\langle x,\xi\rangle} - e^{i\langle x_0,\xi\rangle}\right) \cdot F(\xi)\, d\xi \qquad \forall\, x \in X, \tag{3}$$

where we used the notation $|\xi|_{X,x_0} := \sup_{x\in X} |\langle \xi, x - x_0\rangle|$. We write $\mathcal{B}_{d-1,C}(X, x_0)$ for the class of all such functions.

The functions of Barron class have a bounded first Fourier moment. As we wrote in the introduction, the Barron class provides us an excellent model of high dimensional decision boundaries. Now let us define how they are used to build classifiers:

Definition 2.2. [[9], Definition 2.2] Let $C > 0$ and $d \in \mathbb{N}_{\geq 2}$. Let for $X \subset [0, 1]^{d-1}$ and $x_0 \in X$, $b \in \mathcal{B}_{d-1,C}(X, x_0)$. We call $h_b : X \to \{0, 1\}$ a *horizon function* if :

$$h_b(x) = \mathbb{1}_{b(x^d) \leq x_d}.$$

Now, recall that we want to learn such a horizon function through points i.i.d. sampled from a certain distribution $\mathcal{D}$ over the space. We first formalize what a margin condition on this distribution is.

Definition 2.3. Let $C > 0$, $d \in \mathbb{N}_{\geq 2}$ and $b \in \mathcal{B}_{d-1,C}$. We say that a *$\mu$-margin condition* holds if there is a $\mu > 0$ such that for $M_{\mu,b} := \{x \in [0, 1]^d, \quad |b(x_1, ..., x_{d-1}) - x_d| < \mu\}$, the samples are drawn i.i.d. from a distribution $\mathcal{D}$ on $\mathbb{R}^d$ such that :

$$\mathbb{P}_{x\sim\mathcal{D}}(x \in M_{\mu,b}) = 0.$$

Finally, we define the notion of an α-net :

Definition 2.4. [[16], Definition 3.9] Let $\mathcal{C}$ be a set of bounded real functions and $\alpha > 0$. We say that $N_\alpha \subset \mathcal{C}$ is an α-net of $\mathcal{C}$ for the uniform norm if :

$$\forall f \in \mathcal{C}, \quad \exists f_\alpha \in N_\alpha \quad s.t. \quad ||f - f_\alpha||_{\sup} < \alpha.$$

We now prove that if the margin condition is satisfied for a function h_b, $b \in \mathcal{B}_{d-1,C}(X, x_0)$, then it is guaranteed that for all α smaller or equal than μ, at least one function of from an α-net of $\mathcal{B}_{d-1,C}(X, x_0)$ will perform exactly the same classification as h_b.

Theorem 2.5. *Let h_b a horizon function with its associated boundary function $b \in \mathcal{B}_{d-1,C}(X, x_0)$, $x_0 \in X \subset \mathbb{R}^{d\geq 2}$. If for $\mu > 0$ a μ-margin condition holds with respect to the distribution $\mathcal{D}$ of the data over X, then for all $0 < \alpha \leq \mu$ and N_α an α-net of $\mathcal{B}_{d-1,C}(X, x_0)$, there is a function $\tilde{b} \in N_\alpha$, such that $h_{\tilde{b}} = h_b$ almost surely.*

Proof. Let $b \in \mathcal{B}_{d-1,C}(X, x_0)$ be the boundary function of h_b. Define for $0 < \alpha \leq \mu$:

$$\mathcal{B}_\alpha := \{b_\alpha \in \mathcal{B}_{d-1,C} : ||b_\alpha - b||_\infty < \alpha\}.$$

The set $\mathcal{B}_\mu$ is the set of all the functions of $\mathcal{B}_{d-1,C}(X, x_0)$ that take on $[0,1]^{d-1}$ values that are within the margin $M_{b,\alpha} \subset M_{b,\mu}$.

We first show that : $\forall x \in [0,1]^d \cap M^c_{b,\alpha},\ \forall b_\alpha \in \mathcal{B}_\alpha,\ h_{b_\alpha}(x) = h_b(x)$.

For all points $x \in [0,1]^d \cap M^c_{b,\alpha}$ and all $b_\alpha \in \mathcal{B}_\alpha$, it holds :

$$x \notin \{x \in [0,1]^d, \quad |b(x_1, ..., x_{d-1}) - x_d| \leq \alpha\} \implies b(x_1, ..., x_{d-1}) - x_d \geq \alpha \text{ or } b(x_1, ..., x_{d-1}) - x_d \leq -\alpha.$$

Thus :

$$\begin{cases} b(x_1, ..., x_{d-1}) \geq x_d \implies b(x_1, ..., x_{d-1}) - \alpha \geq x_d, \\ b(x_1, ..., x_{d-1}) \leq x_d \implies b(x_1, ..., x_{d-1}) + \alpha \leq x_d. \end{cases}$$

Moreover, by definition of B_α : $b(x_1, ..., x_{d-1}) - \alpha \leq b_\alpha(x_1, ..., x_{d-1}) \leq b(x_1, ..., x_{d-1}) + \alpha$.

Then :

$$\begin{cases} b(x_1, ..., x_{d-1}) \geq x_d \implies b_\alpha(x_1, ..., x_{d-1}) \geq x_d, \\ b(x_1, ..., x_{d-1}) \leq x_d \implies b_\alpha(x_1, ..., x_{d-1}) \leq x_d. \end{cases}$$

Since $\mathbb{P}_{x\sim\mathcal{D}}(x \notin M_{b,\alpha}) = 1$, these two implications yield :

$$\forall b_\alpha \in \mathcal{B}_\alpha,\ x \sim \mathcal{D} \in [0,1]^d \implies h_{b_\alpha}(x) = h_b(x)\ \ a.s.$$

Now, let N_α be an α-net of $\mathcal{B}_{d-1,C}(X, x_0)$. By definition, there is at least one $\tilde{b} \in N_\alpha$ which is also in B_α.

Therefore, there is a horizon function $h_{\tilde{b}}$ with $\tilde{b} \in N$ such that $h_{\tilde{b}} = h_b$ on $[0,1]^d$ almost surely. □

Now that we have proven that a subset of $\mathcal{B}_{d-1,C}$ suffices as a hypothesis set in order to be in a realizable case, we will use the fact that it is a finite set to deduce the corresponding estimation rate.

Definition 2.6. [[16], Definition 3.9] Let α be a positive real number. We call *α-covering entropy* of a set K the number:

$$M_K(\alpha) = \ln(V_K(\alpha)) \quad \text{where} \quad V_K(\alpha) := \min\{|G| : G \text{ is an } \alpha\text{-net of K}\}.$$

Proposition 2.7. [[9], Proposition 4.4] Given $d \in \mathbb{N}$ and $C > 0$, there exists a constant $C_0 = C_0(d, C) > 0$ such that the covering entropy numbers $M_{\mathcal{B}_{d,C}}$ of $\mathcal{B}_{d,C}(X, x_0)$ with respect to the uniform norm on $[0,1]^d$ satisfy

$$M_{\mathcal{B}_{d,C}}(\alpha) \leq C_0 \cdot \alpha^{-1/(\frac{1}{2}+\frac{1}{d})} \cdot (1 + \ln(1/\alpha)).$$

There is a finite subset of the Barron class which suffices as hypothesis set to achieve a perfect classification. We now prove the corresponding bounds in statistical learning. We first introduce some definitions before stating the Proposition 2.9 from which we compute the bounds.

Definition 2.8. [[18], 2.2 and 3.2.2] Let $\mathfrak{A}$ be an algorithm aiming at estimating a classifier h from a sample S of size m drawn i.i.d. according to a distribution $\mathcal{D}$. We define the *empirical risk* $\hat{\mathcal{R}}^h_S$ of the algorithm $\mathfrak{A}(S)$ as

$$\hat{\mathcal{R}}^h_S(\mathfrak{A}(S)) = \frac{1}{m}\sum_{x\in S} \mathbb{1}_{\mathfrak{A}(S)(x)\neq h(x)}$$

and its *generalization error*, *risk* or *true error* $\mathcal{R}$ as

$$\mathcal{R}^h_{\mathcal{D}}(\mathfrak{A}(S)) = \mathbb{E}_{S'\sim\mathcal{D}^m}(\hat{\mathcal{R}}^h_{S'}(\mathfrak{A}(S))) = \mathbb{P}_{x\sim\mathcal{D}}(\mathfrak{A}(S)(x) \neq h(x)).$$

Proposition 2.9. [[18], Corollary 2.3] Let $\mathcal{H}$ be the hypothesis class and $\mathcal{C} \subset \mathcal{H}$ be the concept class. Let $\mathfrak{A}$ be an algorithm aiming at learning $c \in \mathcal{C}$, such that for each $h \in \mathcal{H}$, and each sample $S = (x_i, h(x_i))_{i=1}^n$ drawn i.i.d. according to a distribution $\mathcal{D}$ we have that :

$$\hat{\mathcal{R}}^c_S(\mathfrak{A}(S)) = 0.$$

Then, for every $\delta > 0$, with probability $1 - \delta$ over the sampling of S,

$$\mathcal{R}^c_{\mathcal{D}}(\mathfrak{A}(S)) \leq \frac{1}{n}\left(\log|\mathcal{H}| + \log\frac{1}{\delta}\right).$$

Now we will combine the above proposition together with Theorem 2.5 and Proposition 2.7. We saw that if one can manage to have a classifier inferred from a sample of size $n > 0$ with an empirical error on this sample equals to zero and a finite hypothesis set, then the true error of this classifier will decay with $\frac{1}{n}$. Theorem 2.5 ensures that if a margin condition holds, then any hypothesis class containing a net tighter than the margin contains at least a perfect classifier as well. It contains therefore *a fortiori* at least one classifier with a zero empirical error on any sample. Moreover, thanks to Proposition 2.7, this hypothesis set is finite. We can then apply Proposition 2.9 :

Theorem 2.10. *For $d \geq 2$ and $C \in \mathbb{R}^*$, let $b \in \mathcal{B}_{d-1,C}(X, x_0)$ and h_b the associated horizon function. Suppose that for $\mu > 0$ a μ-margin condition holds w.r.t. the distribution $\mathcal{D}$ of the data over $[0,1]^d$. If for $0 < \alpha \leq \mu$ the hypothesis class $\mathcal{H}$ contains an α-net N_α of $\mathcal{B}_{d-1,C}(X, x_0)$ and is finite, then there exists an algorithm $\mathfrak{A}$ estimating h_b from a sample S of size n drawn i.i.d. according to the distribution $\mathcal{D}$ such that $\forall \delta > 0$:*

$$\mathcal{R}(\mathfrak{A}(S)) \leq \frac{1}{n}\left(M_{\mathcal{B}_{d-1,C}}(\alpha) + \log\frac{1}{\delta}\right),$$

with probability $1 - \delta$.

Proof. Since a μ-margin condition holds, Theorem 2.5 tells that for all $0 < \alpha \leq \mu$, any α-net N_α of $\mathcal{B}_{d-1,C}(X, x_0)$ contains at least one horizon function $\hat{b}$ of a classifier that makes no error. Moreover, Proposition 2.7 tells that this net is finite, with size $M_{\mathcal{B}_{d-1,C}}(\alpha) \leq C_0 \cdot \alpha^{-1/(\frac{1}{2}+\frac{1}{d-1})} \cdot (1 + \ln(1/\alpha))$, C_0 a constant. Therefore, if for $0 < \alpha \leq \mu$ and N_α such a net, $\mathcal{H} \supset N_\alpha$, then there exists at least one algorithm $\mathfrak{A}$ such that :

$$\hat{\mathcal{R}}_S(\mathfrak{A}(S)) = 0.$$

Indeed, you can always try out all the hypothesis set $\mathcal{H}$ in the worst case.

We conclude using Proposition 2.9.

□

Thus, we have an excellent risk rate in this configuration. Now, the question is whether we can generalize this bound to sets that are only locally with Barron boundary and with neural networks. Such a rate does not seem reachable with no more assumptions as a set of neural network with fixed size is not finite. Still, we can compute some good bounds as even sets only locally Barron can be approximated very well by neural networks as we prove in the next section.

3 Approximation of a Barron-bounded Set by a Neural Network

In this section, we study the question whether the rate $\frac{1}{n}$, n being the sample size, or something similar is achievable with neural networks. The issue here is that the size of a neural network class is not finite. Therefore, using a set of neural network as hypothesis set is not compatible with the rate computed in 2.10. However, while of infinite size, a class of neural networks with fixed numbers of layers and neurons has a bounded VC-dimension. The *Fundamental Theorem of Statistical Learning* found in [18] (Theorem 6.8) provides us with some bounds on the risk of an estimator following an ERM rule if the hypothesis class to which the estimator belongs is of a bounded VC-dimension. We first state the *Fundamental Theorem of Statistical Learning* :

Theorem 3.1. *[[18], Theorem 6.8] Let $\mathcal{H}$ be a hypothesis class of functions from a domain X to $\{0,1\}$. Let $\mathcal{C}$ be the concept class and $c \in \mathcal{C}$ the classifier to learn. Assume that $VCdim(\mathcal{H}) < \infty$ and that for some distribution $\mathcal{D}$ over X it holds that : $\min_{h\in\mathcal{H}} \mathcal{R}^c_{\mathcal{D}}(h) = 0$. Then, there is an absolute constant $K_1 \in \mathbb{R}$ such that with probability $1-\delta$ over the sampling of $S = (x_1,\ \dots\ , x_n) \sim \mathcal{D}^n$:*

$$\forall S \sim \mathcal{D}^n,\ \exists h_S \in \mathcal{H} \text{ s.t. } \hat{\mathcal{R}}^c_S(h_S) = 0 \text{ and } \mathcal{R}^c_{\mathcal{D}}(h_S) \leq \frac{\log(n)}{n} K_1\, VCdim(\mathcal{H}) + \frac{K_1}{n}\log(1/\delta).$$

If one can prove that there exists a class of neural networks $\mathcal{H}$ such that for a certain distribution $\mathcal{D}$ over $X \subset \mathbb{R}^d$ for which a margin condition holds, there is at least one classifier in $\mathcal{H}$ that makes no error on a n-sized sample, then the true risk of this classifier would be equivalent to $\frac{\log(n)}{n}$, which is a highly satisfying rate. To this end, we prove in this section that neural networks can approximate well the characteristic function of sets with Barron boundary. However, we won't stick to the sole sets with a single function of the Barron class as boundary, rather we prove that neural networks can estimate the characteristic function of sets in $\mathbb{R}^d$ that can be approximated like sets with piece-wise Barron boundary, not necessarily continuous (see Definition 3.4). We rely on the following Propsition 3.2 to prove that such a class of neural network set exists.

Proposition 3.2. [[9], Proposition 2.2] There is a universal constant $\kappa > 0$ with the following property: For any bounded set $X \subset \mathbb{R}^d$ with nonempty interior, for all $C \in \mathbb{R}^*_+$, $x_0 \in X$ and $f \in \mathcal{B}_{d,C}(X, x_0)$, and all $N \in \mathbb{N}$, there is a shallow neural network Φ with $8N$ neurons in the hidden layer such that

$$\|f - R_\sigma \Phi\|_{\sup} \leq \kappa \sqrt{d} \cdot C \cdot N^{-1/2}.$$

Furthermore, one can choose all weights and biases of Φ to be bounded by

$$\big(5 + \vartheta(X, x_0)\big) \cdot \big(1 + \|x_0\|_1\big) \cdot \sqrt{C}, \quad \text{where} \quad \vartheta(X, x_0) := \sup_{\xi \in \mathbb{R}^d \setminus \{0\}} \Big(\|\xi\|_\infty / |\xi|_{X,x_0}\Big).$$

The proof, given in appendix, relies on the fact that due to their boundedness Barron functions can be rewritten as bounded expectation of half-spaces. This expectation of half-spaces is itself the realization of a certain neural network, the parameters being distributed according to a distribution deriving from the function. From this equality one can conclude the boundedness of the expectation of the norm of the difference of the function and the neural network. Since the expectation is bounded, there exists at least one realization for the parameters such that the norm of the difference itself is bounded, which concludes the proof.

Remark : The quantity $\vartheta(X, x_0)$ roughly speaking measures how big of a rectangle the set X contains. More precisely, assume that $X \supset [a, b]$ where $b_i - a_i \geq \varepsilon > 0$ for all $i \in d$. Then we see with the standard basis $(e_1, \dots, e_d)$ of $\mathbb{R}^d$ that

$$\varepsilon\,|\xi_i| = \big|\langle \xi, a + \varepsilon\, e_i - x_0\rangle - \langle \xi, a - x_0\rangle\big| \leq |\langle \xi, a +_i - x_0\rangle| + |\langle \xi, a - x_0\rangle| \leq 2 \sup_{x\in X} |\langle \xi, x - x_0\rangle|.$$

Since this holds for all $i \in d$, we see $|\xi|_{X,x_0} \geq \frac{\varepsilon}{2}\,\|\xi\|_\infty$ and hence $\vartheta(X, x_0) \leq \frac{2}{\varepsilon}$. Note that since X has nonempty interior, we can always find a sufficiently small non-degenerate rectangle in X; therefore, $|\xi|_{X,x_0} \gtrsim \|\xi\|_{\ell^\infty}$ for all $\xi \in \mathbb{R}^d$.

Now, for the sake of generality, we introduce the *Barron approximation space*, the space of functions that can be approximated by a neural network like in Proposition 3.2.

Definition 3.3. [[9], Definition 3.1] Let $d \geq 2$ and let $X \subset \mathbb{R}^d$ be bounded with nonempty interior. For $C \in \mathbb{R}_+^*$, we define the *Barron approximation set* $\mathcal{BA}_{d,C}(X)$ as the set of all functions $f : X \to \mathbb{R}$ such that for every $N \in \mathbb{N}$ there is a shallow neural network Φ with N neurons in the hidden layer such that

$$\|f - R_\sigma \Phi\|_{\sup} \leq \sqrt{d} \cdot C \cdot N^{-1/2}$$

and such that all weights (and biases) of Φ are bounded in absolute value by

$$\sqrt{C} \cdot \Big(5 +_{x_0 \in X} \big[\, \|x_0\|_1 + \vartheta(X, x_0) \,\big]\Big), \quad \text{where} \quad \vartheta(X, x_0) := \sup_{\xi \in \mathbb{R}^d \setminus \{0\}} \big(\|\xi\|_\infty / |\xi|_{X,x_0}\big).$$

The set $\mathcal{BA}_d(X) = \bigcup_{C \in \mathbb{R}_+^*} \mathcal{BA}_{d,C}(X)$ is called the *Barron approximation space.*

Remark : Thanks to Proposition 3.2, there is an absolute constant κ such that for all $X \subset \mathbb{R}^d$ and all $x_0 \in X$, the Barron class $\mathcal{B}_{d,\kappa C}(X, x_0)$ is included in $\mathcal{BA}(X)$.

Definition 3.4. [[9], Definition 3.3] Let $d \geq 2$ and $C \in \mathbb{R}_+^*$ and let $Q = [a, b] \subset \mathbb{R}^d$ be a rectangle. A function $h_b : Q \to \mathbb{R}$ is called a *Barron horizon function with constant* C, if there are $i \in [\![1, d]\!]$ and $h \in \mathcal{BA}_{d-1,C}\big(Q^i\big)$ where $Q^i = [a^i, b^i]$ as well as $\theta \in \{\pm 1\}$ such that

$$h_b(x) = \mathbb{1}_{\theta x_i \leq b(x^i)} \qquad \forall\, x \in Q.$$

We write $\mathcal{BH}_{d,C}(Q)$ for the set of all such functions.

Finally, given $M \in \mathbb{N}$ and $C \in \mathbb{R}_+$, a compact set $\Omega \subset \mathbb{R}^d$ is said to have a *Barron class boundary with constant* B if there exist rectangles $Q_1, \ldots, Q_M \subset \mathbb{R}^d$ such that $\Omega \subset \bigcup_{i=1}^M Q_i$ where the rectangles have disjoint interiors (i.e., $Q_i^\circ \cap Q_j^\circ = \emptyset$ for $i \neq j$) and such that $\mathbb{1}_{Q_i \cap \Omega} \in \mathcal{BH}_{d,C}(Q_i)$ for each $i \in M$. We write $\mathcal{BB}_{C,M}(\mathbb{R}^d)$ for the class of all such sets. Also, a family $(Q_j)_{j=1}^M$ of rectangles as above is called an *associated cover* of Ω.

Definition 3.5. Let $X \subset \mathbb{R}^{d \geq 2}$ be a set. We call *boundary* of X the set $\partial X := \{x \in \mathbb{R}^d : \forall \nu > 0, \exists y, y' \in B(x, \nu), y \in X, y' \notin X\}$ where $B(x, \nu) := \{y \in \mathbb{R}^d : ||x - y||_2 < \nu\}$.

Remark : For $d \geq 2$ and $C \in \mathbb{R}^*$, let $b \in \mathcal{B}_{d,C}$ and h_b the associated horizon function. The set $X = \{x \in \mathbb{R}^d : h_b(x) = 1\}$ has the following boundary $\partial X = \{x \in \mathbb{R}^d : b(x^d) = x_d\}$.

Definition 3.6. Let $X \subset \mathbb{R}^{d \geq 2}$ be a set. For $x \in \mathbb{R}^d$, we call *distance of x from X* the value $||x - X|| := \inf\{||x - y||_2 : y \in X\}$.

We said at the beginning of this section that its goal was to prove that neural networks could learn well the characteristic function of sets with piece-wise Barron boundary. To be more precise, what is meant with "learn well" is that the classifier obtained differs from the actual characteristic function on an area whose size decreases along with the increase of the neural network size. Moreover, we prove that this area corresponds to a tube of width of order $\frac{1}{\sqrt{N}}$, with the neural network having a number of neurons linearly linked to some $N > 0$.

Theorem 3.7. *Let $d \geq 2$ and $\Omega \subset \mathbb{R}^d$ such that for $M > 0$ and $C \in \mathbb{R}_+^*$, $\Omega \in \mathcal{BB}_{C,M}(\mathbb{R}^d)$. There exists a neural network I_N with three hidden layers and the ReLU activation function σ such that :*

$$\forall x \in \mathbb{R}^d, \quad ||x - \partial\Omega|| > 3\gamma N^{-1/2} \implies \mathbb{1}_\Omega(x) = R_\sigma I_N(x), \tag{4}$$

where $\gamma := C\sqrt{d-1}$.

Moreover, $0 \leq R_\sigma I_N(x) \leq 1$ for all $x \in \mathbb{R}^d$ and the architecture of I_N is given by

$$\mathcal{A} = (d, M(N + 2d + 2), M(4d + 2), M, 1, 1).$$

Thus, I_N has at most $7M(N + d)$ neurons and at most $54\ d^2\ M\ N$ non-zero weights. The weights (and biases) of I_N are bounded in magnitude by $d(4 + R)(1 + C) + \sqrt{N}(C^{-1} + C^{-1/2})$, where $R = \sup_{x \in \Omega} ||x||_\infty$.

Proof. The proof consists in four steps that could be gathered together in two main parts. We first approximate locally Ω on every cubes of its associated cover (steps 1, 2 and 3), then we build a network from the local ones (step 4). Before starting, we want to reformulate a bit the assumptions made on Ω.

Let $(Q_m)_{m=1}^M$ be an associated cover of Ω : $Q_m = [a_m, b_m]$ with $a_m, b_m \in \mathbb{R}^d$.

For some $\varepsilon > 0$ let us construct $(\tilde{Q}_m)_{m=1}^M$ where :

$$(\tilde{Q}_m) := \Pi_{j=1}^d [a_{m,j} - \varepsilon, b_{m,j} + \varepsilon].$$

By the assumption that $\Omega \in \mathcal{BB}_{C,M}(\mathbb{R}^d)$, there exist $(f_m)_{m=1}^M \subset \mathcal{BA}_C(\mathbb{R}^{d-1})$, $i \in [\![1, d]\!]$ and $\theta_m \in \{-1, 1\}$ such that :

$$\begin{cases} \forall x \in Q_m, \quad \mathbb{1}_\Omega(x) = \mathbb{1}_{f_m(x^i) \geq \theta_m x_i}, \\ \forall x \in Q_m, \quad \forall y \in \tilde{Q}_m \setminus Q_m, \quad ||x^i - y^i||_2 < \varepsilon \implies |f_m(x_i) - f_m(y_i)| < \varepsilon. \end{cases}$$

Now we can construct a neural network that approximates $\mathbb{1}_\Omega$ by approximating successively the $(\mathbb{1}_{\Omega \cap Q_m})_{m=1}^M$.

Step 1 : local approximation of the $(f_m)_{m=1}^M$. As we said in Definition 3.4, there exists for every Q_m a function $f_m \in \mathcal{BA}_C(\mathbb{R}^{d-1})$, $i \in [\![1, d]\!]$ and $\theta_m \in \{-1, 1\}$ such that $\mathbb{1}_{\Omega \cap Q_m} = \mathbb{1}_{f_m(x^i) \geq \theta_m x_i}$. From 3.3, there is as well a shallow neural network I_N^m with N neurons in the hidden layer such that $||f_m - R_\sigma I_N^m||_{\text{sup}} \leq \gamma N^{-1/2}$ where $\gamma := C\sqrt{d-1}$. The weights and biases of I_n^m are bounded by $\sqrt{C} \cdot \left(6 + \vartheta(Q_m^i, q_m) + ||q_m||_1\right)$, $q_m \in Q_m^i$.

Step 2 : approximation of the horizon functions $h_m(x) = \mathbb{1}_{\theta_m x_i \leq f_m(x^i)}$. Denoting :

i) $S_m := \{x \in \tilde{Q}_m : f_m(x^i) \geq \theta_m x_i\}$,

ii) $T_m := \{x \in \tilde{Q}_m : R_\sigma I_N^m(x^i) \geq \theta_m x_i\}$, it can easily be shown that :

$$S_m \triangle T_m \subset \{x \in \tilde{Q}_m : |f_m(x^i) - \theta_m x_i| \leq \gamma N^{-1/2}\}.$$

This implies :

$$\{x \in \tilde{Q}_m : h_m(x) \neq \mathbb{1}_{T_m}(x)\} \subset \{x \in \tilde{Q}_m : |f_m(x^i) - \theta_m x_i| \leq \gamma N^{-1/2}\}.$$

Next, for $\tau > 0$ define the ReLU approximated Heaviside function :

$$H_\tau(x) = \tau^{-1}(\sigma(x) - \sigma(x - \tau)).$$

We can compose this function with $R_\sigma I_N^m(x^i) - \theta_m x_i$ to approximate $\mathbb{1}_{T_m}(x)$.

Defining $R_\sigma J_N^m(x) := H_{\gamma N^{-1/2}}(R_\sigma I_N^m(x^i) - \theta_m x_i)$, we have :

$$\{x \in \tilde{Q}_m : \mathbb{1}_{T_m}(x) \neq R_\sigma J_N^m(x)\} \subset \{x \in \tilde{Q}_m : |R_\sigma I_N^m(x^i) - \theta_m x_i| \leq \gamma N^{-1/2}\}.$$

Moreover :

$$\{x \in \tilde{Q}_m : h_m(x) \neq R_\sigma J_N^m(x)\} \subset \{x \in \tilde{Q}_m : \mathbb{1}_{T_m}(x) \neq R_\sigma J_N^m(x)\} \cup \{x \in \tilde{Q}_m : \mathbb{1}_{T_m}(x) \neq h_m(x)\}.$$

Thus :

$$\{x \in \tilde{Q}_m : h_m(x) \neq R_\sigma J_N^m(x)\} \subset \{x \in \tilde{Q}_m : |f_m(x^i) - R_\sigma I_N^m(x^i)| + |R_\sigma I_N^m(x^i) - \theta_m x_i| \leq 2\gamma N^{-1/2}\}.$$

Now, using the fact that for $x \in \tilde{Q}_m$, $\quad |f_m(x^i) - \theta_m x_i| \leq |f_m(x^i) - R_\sigma I_N^m(x^i)| + |R_\sigma I_N^m(x^i) - \theta_m x_i|$, we can conclude that :

$$\{x \in \tilde{Q}_m : h_m(x) \neq R_\sigma J_N^m(x)\} \subset \{x \in \tilde{Q}_m : |f_m(x^i) - \theta_m x_i| \leq 2\gamma N^{-1/2}\}. \tag{5}$$

Remark : This means that on a Q_m, the neural network performs its error only within a $2\gamma N^{-1/2}$-margin around the actual Barron boundary.

Step 3 : restricting to Q_m. Here, we want to construct an approximation of $x \to \mathbb{1}_{Q_m}(x) R_\sigma J_N^m(x)$ with the help of ReLU functions only. For $j \in [\![1, d]\!]$, let $t_j : \mathbb{R} \to [0, 1]$ be

$$t_j(u) := \begin{cases} 0 & \text{if } u \in \mathbb{R} \setminus [a_j - \varepsilon, b_j + \varepsilon], \\ 1 & \text{if } u \in [a_j, b_j], \\ \dfrac{u - a_j}{\varepsilon} & \text{if } u \in [a_j - \varepsilon, a_j], \\ \dfrac{b_j - u}{\varepsilon} & \text{if } u \in [b_j, b_j + \varepsilon]. \end{cases}$$

and

$$\eta_\varepsilon(x, y) : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R} := \sigma\left(\sigma(y) + \sum_{j=1}^{d} t_j(x_j) - d\right).$$

Now we can build the neural network L_N^m s.t. $R_\sigma L_N^m(x) = \eta_\varepsilon(x, R_\sigma J_N^m(x))$ and note that :

$$\{x \in \mathbb{R}^d : \mathbb{1}_{Q_m}(x) R_\sigma J_N^m(x) \neq R_\sigma L_N^m(x)\} \subset \tilde{Q}_m \setminus Q_m,$$

as well as :

$$\text{supp} R_\sigma L_N^m \subset \text{supp} R_\sigma J_N^m \subset \tilde{Q}_m, \tag{6}$$

since $R_\sigma J_N^m(x) = 0 \implies R_\sigma L_N^m(x) = 0$.

To conclude this step, note that L_N^m yields the same result as J_N^m (which is the approximation of h_m) on Q_m and vanishes outside $\tilde{Q}_m$. The challenge now is to build a single overall network we want as close as possible to L_N^m on $\tilde{Q}_m$.

Step 4 : concatenation of the networks. Consider the smoothed sum of the $(L_N^m)_{m=1}^M$:

$$R_\sigma I_N(x) = \sigma\left(\sum_{m=1}^{M} R_\sigma L_N^m(x)\right).$$

Note that for $m \leq M,\ x \in Q_m$, we have $R_\sigma I_N(x) \geq R_\sigma L_N^m(x)$.

Let $m \leq M$. We now show that I_N error area is included in the following tube around the boundary of Ω :

$$\{x \in Q_m : h_m(x) = \mathbb{1}_{\theta_m x_i \leq f_m(x^i)} \neq R_\sigma I_N(x)\} \subset \{x \in Q_m : ||x - \partial\Omega|| \leq 3\gamma N^{-1/2}\}$$

It holds that :

$$\{x \in Q_m : h_m(x) \neq R_\sigma I_N(x)\} \subset \{x \in Q_m : h_m(x) \neq R_\sigma L_N^m(x)\} \cup \{x \in Q_m : R_\sigma L_N^m(x) \neq R_\sigma I_N(x)\}.$$

We denote the first member of this union $Q_m^{L \neq h} := \{x \in Q_m : h_m(x) \neq R_\sigma L_N^m(x)\}$ and the second one $Q_m^{L \neq I} := \{x \in Q_m : R_\sigma L_N^m(x) \neq R_\sigma I_N(x)\}$.

We will now prove that on $x \in Q_m^{L \neq h} \setminus Q_m^{L \neq I}$ as well as on $Q_m^{L \neq I}$, the network I_N makes errors only if evaluated on points that are at a distance smaller than $3\gamma N^{-1/2}$ from the boundary of Ω.

i) If $x \in Q_m^{L \neq h} \setminus Q_m^{L \neq I}$, we have by definition:

$$R_\sigma I_N(x) = R_\sigma L_N^m(x).$$

But Step 3 lead to :

$$\forall x \in Q_m, \quad R_\sigma L_N^m(x) = R_\sigma J_N^m(x).$$

Therefore for $x \in Q_m^{L \neq h} \setminus Q_m^{L \neq I}$, we have $R_\sigma I_N(x) = R_\sigma J_N^m(x)$ and (5) yields :

$$\{x \in Q_m^{L \neq h} \setminus Q_m^{L \neq I} : R_\sigma I_N^m(x) \neq h_m(x)\} \subset \{x \in Q_m : |f_m(x^i) - \theta_m x_i| \leq 2\gamma N^{-1/2}\}.$$

This can be reformulated into :

$$\forall x \in Q_m^{L \neq h} \setminus Q_m^{L \neq I},\ R_\sigma I_N(x) \neq h_m(x) \implies ||x - \partial\Omega|| \leq 2\gamma N^{-1/2}.$$

ii) Otherwise, if $x \in Q_m^{L \neq I}$, we have :

$$R_\sigma I_N(x) > R_\sigma L_N^m(x) = R_\sigma J_N^m(x).$$

But remark that $R_\sigma I_N(x) - R_\sigma J_N^m(x) \leq \sum_{k=1, k\neq m}^{M} R_\sigma L_N^k(x)$. Then

$$R_\sigma I_N(x) - R_\sigma J_N^m(x) > 0 \implies \exists k \in [\![1, M]\!] \setminus \{m\},\ R_\sigma L_N^k(x) > 0.$$

Therefore, we have that if $x \in Q_m^{L \neq I}$, then $x \in \text{supp} R_\sigma L_N^k$. But from 6 we know that $\text{supp} R_\sigma L_N^k \subset \text{supp} R_\sigma J_N^k$. From 5 we also know that $\text{supp} R_\sigma J_N^k \subset \text{supp} h_k(x) \cup \{x \in \tilde{Q}_k : |f_k(x^l) - \theta_k x_l| \leq 2\gamma N^{-1/2}\}$, for some $l \in [\![1, M]\!]$. Thus:

$$\forall x \in \tilde{Q}_k, \quad x \in \text{supp} R_\sigma L_N^k \implies ||x - \tilde{Q}_k \cap \Omega|| \leq 2\gamma N^{-1/2}.$$

We infer from this that : $\forall x \in Q_k, \quad x \in \text{supp} R_\sigma L_N^k \implies ||x - Q_k \cap \Omega|| \leq 2\gamma N^{-1/2} + \varepsilon.$

Moreover, since the cubes are disjoint, we know that the points of $Q_m^{L \neq h}$ in $\text{supp} R_\sigma L_N^k$, while close to it, cannot be in $Q_k \cap \Omega$:

$$x \in Q_m^{L \neq I} \implies \exists k \in [\![1, M]\!] \setminus \{m\},\ ||x - \partial(Q_k \cap \Omega)|| \leq \ 2\gamma N^{-1/2} + \varepsilon. \tag{7}$$

The last problem is that points can be close to the boundary of $Q_k \cap \Omega$, but "deep inside" $Q_m \cap \Omega$. $\partial(Q_k \cap \Omega)$ being only included in $\partial\Omega$ but not equal to it, we have to fix this problem. For $x \in Q_m$:

$$R_\sigma J_N^m(x) = 1 \implies R_\sigma J_N^m(x) = R_\sigma L_N^m(x) = R_\sigma I_N(x).$$

Thus :

$$R_\sigma L_N^m(x) \neq R_\sigma I_N(x) \implies R_\sigma J_N^m(x) < 1.$$

From the latter follows :

$$Q_m^{L \neq I} \subset \{x \in Q_m : |f_m(x^i) - \theta_m x_i| \leq 2\gamma N^{-1/2}\} \cup \Omega^c.$$

So if we restrict $Q_m^{L \neq I}$ to Ω, points of this set cannot be at a distance from $\partial\Omega$ greater than $2\gamma N^{-1/2}$:

$$\forall x \in Q_m^{L \neq I} \cap \Omega,\ ||x - \partial\Omega|| \leq 2\gamma N^{-1/2}.$$

Moreover, from 7 : $x \in Q_m^{L \neq I} \cap \Omega^c \implies ||x - \partial\Omega|| \leq 2\gamma N^{-1/2} + \varepsilon.$

Finally :

$$\forall x \in Q_m^{L \neq I},\ R_\sigma I_N(x) \neq h_m(x) \implies ||x - \partial\Omega|| \leq 2\gamma N^{-1/2} + \varepsilon.$$

Setting $\varepsilon = \gamma N^{-1/2}$, and recalling that $\{x \in Q_m :\ R_\sigma I_N(x) \neq h_m(x)\} = Q_m^{L \neq h} \cup\ Q_m^{L \neq I}$:

$$\forall x \in Q_m,\ R_\sigma I_N(x) \neq h_m(x) \implies ||x - \partial\Omega|| \leq 3\gamma N^{-1/2}.$$

Moreover, $\Omega \subset \cup_{m=1}^N Q_m$, therefore if $x \notin \cup_{m=1}^N Q_m$, then $\mathbb{1}_\Omega(x) = 0$. So, if $R_\sigma I_N(x) \neq \mathbb{1}_\Omega(x)$ outside the union of the $(Q_m)_{m=1}^M$, then it means that $R_\sigma I_N(x) > 0$ and hence that for some $m \in [\![1, M]\!]$, $R_\sigma L_N^m(x) > 0$. This can happen only at a distance at most $2\gamma N^{-1/2} + \varepsilon = 3\gamma N^{-1/2}$ from $\partial\Omega$. We can conclude the proof :

$$\forall x \in \mathbb{R}^d,\ R_\sigma I_N(x) \neq h_m(x) \implies ||x - \partial\Omega|| \leq 3\gamma N^{-1/2}.$$

□

4 Related Estimation Bounds

We are able to quantify and bound an area around the boundary of a set where neural networks learning the indicator function of said set will fail. In this section, we link this result to the margin condition described in Section 2. Indeed, if we know that such a margin condition holds and have a neural network that makes errors exclusively within the margin (this neural network exists, as we stated it in Theorem 3.7), we have thereby a classifier that makes no error at all. The VC-dimension of a set of neural networks with bounded numbers of layers and neurons being in addition bounded (from Theorem 2.1 in [6]), we can apply Theorem 3.1 to compute some bounds for this classifier risk.

We first need to make some slight technical adjustments before stating the theorem, just in order to suit the framework of [6], where classifiers do not output into $\{0,1\}$ but rather into $\{-1,1\}$:

Definition 4.1. We define the following function $\mathrm{sign} : \mathbb{R} \to \{-1,1\}$, $\mathrm{sign}(x) = -1$ if $x \leq 0$ and $\mathrm{sign}(x) = 1$ if $x > 0$.

Definition 4.2. We also define the following function for $\Omega \subset \mathbb{R}^d$:

$$\chi_\Omega : \mathbb{R}^d \to \{-1,1\},\ \chi_\Omega(x) = \begin{cases} -1 & \text{if } x \notin \Omega, \\ 1 & \text{if } x \in \Omega. \end{cases}$$

Theorem 4.3. *Let $d \geq 2$ and $\Omega \subset \mathbb{R}^d$ such that for $M > 0$ and $C \in \mathbb{R}_+^*$, $\Omega \in \mathcal{BB}_{C,M}(\mathbb{R}^d)$. Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d$. Suppose that for $\mu > 0$, a margin condition holds :*

$$\mathbb{P}_{x\sim\mathcal{D}}\left(x \in \{y \in \mathbb{R}^d : ||x - \partial\Omega|| > \mu\}\right) = 1. \tag{8}$$

Then, there exist $K_1, K_2 \in \mathbb{R}$, absolute constants, and a class of ReLU neural networks $\mathcal{N}(\mathcal{A})$ with architecture detailed below such that, for $n > 0$, with a probability $1 - \delta$ regarding the sampling of $S = (x_1, \dots, x_n) \sim \mathcal{D}^n$:

$$\forall S \sim \mathcal{D}^n, \quad \exists \Phi_S^* \in \mathcal{N}(\mathcal{A}) \ s.t.\ \#\{x \in S : \mathrm{sign}(R_\sigma \Phi_S^*(x)) \neq \chi_\Omega(x)\} = 0, \tag{9}$$

and :

$$\mathbb{P}_{x\sim\mathcal{D}}(\mathrm{sign}(R_\sigma \Phi_S^*(x)) \neq \chi_\Omega(x)) \leq \frac{\log(n)}{n} K_1 K_2 M^2 d^2 N \log(dMN) + \frac{K_1}{n}\log(1/\delta). \tag{10}$$

Or, in a summarized way :

$$\forall S \sim \mathcal{D}^n, \exists \Phi_S^* \in \mathcal{N}(\mathcal{A}),\ s.t.\ \hat{\mathcal{R}}_S^{\chi_\Omega}(\mathrm{sign}(R_\sigma \Phi_S^*)) = 0 \ and\ \mathcal{R}_{\mathcal{D}}^{\chi_\Omega}(\mathrm{sign}(R_\sigma \Phi_S^*)) \leq \frac{\log(n)}{n}\mathcal{O}(\log(d)d^2). \tag{11}$$

The architecture $\mathcal{A}$ of the networks of $\mathcal{N}(\mathcal{A})$ is given by

$$\mathcal{A} = (d, M(N + 2d + 2), M(4d + 2), M, 1, 1),$$

where $N > \left(\frac{3\gamma}{\mu}\right)^2$ with $\gamma := C\sqrt{d-1}$.

Thus, I_N has at most $7M(N + d)$ neurons and at most $54\ d^2\ M\ N$ non-zero weights. The weights (and biases) of I_N are bounded in magnitude by $d(4 + R)(1 + C) + \sqrt{N}(C^{-1} + C^{-1/2})$, where $R = \sup_{x\in\Omega} ||x||_\infty$.

Remark : This is the best rate possible without any further assumption.

Proof. This theorem is again a direct consequence of the "Fundamental Theorem of Statistical Learning" found in [18] and stated in this thesis as Theorem 3.1. We first have to prove that if a margin condition holds for $\mathcal{D}$, the class $\mathcal{N}(\mathcal{A})$ fulfills the condition of containing a classifier with a zero risk. We made this assumption : $\mathbb{P}_{x\sim\mathcal{D}}\left(x \in \{y \in \mathbb{R}^d : ||x - \partial\Omega|| > \mu\}\right) = 1$. Moreover, Theorem 3.7 tells that for $N > \left(\frac{3\gamma}{\mu}\right)^2$, there exists a neural network with architecture $\mathcal{A} = (d, M(N+2d+2), M(4d+2), M, 1, 1)$ such that :

$$\forall x \in \mathbb{R}^d, \quad ||x - \partial\Omega|| > \mu \implies \mathbb{1}_\Omega(x) = R_\sigma I_N(x).$$

So : $\mathbb{P}_{x\sim\mathcal{D}}\left(\mathbb{1}_\Omega(x) = R_\sigma I_N(x)\right) \geq \mathbb{P}_{x\sim\mathcal{D}}\left(x \in \{y \in \mathbb{R}^d : ||x - \partial\Omega|| > \mu\}\right) = 1.$

We conclude that : $\mathbb{P}_{x\sim\mathcal{D}}\left(\mathbb{1}_{\Omega}(x) = R_\sigma I_N(x)\right) = 1$, which is equivalent to $\mathcal{R}_{\mathcal{D}}^{\mathbb{1}_\Omega}(R_\sigma I_N) = 0$.

We now build the following neural network $\Phi^* \in \mathcal{N}(\mathcal{A})$, such that : $R_\sigma \Phi^*(x) = R_\sigma I_N(x) - \frac{1}{2}$.

Note that if $R_\sigma I_N(x) = \mathbb{1}_\Omega(x)$, then $\text{sign}(R_\sigma \Phi^*(x)) = \chi_\Omega(x)$. Therefore :

$$\mathbb{P}_{x\sim\mathcal{D}}\left(\text{sign}(R_\sigma\Phi^*(x)) = \chi_\Omega(x)\right) = 1 \text{ and } \mathcal{R}_{\mathcal{D}}^{\chi_\Omega}(\text{sign}(R_\sigma\Phi^*)) = 0.$$

We set $\mathcal{H} := \{\text{sign} \circ R_\sigma\Phi : \Phi \in \mathcal{N}(\mathcal{A}).\}$ Thus, there exists a classifier $h = \chi \circ R_\sigma I_N \in \mathcal{H}$ with a zero risk (and therefore a empirical error null as well). Moreover, Theorem 2.1 in [6] tells us that the VC dimension of $\mathcal{H}$ is bounded as follows :

$$\exists K_2 \in \mathbb{R},\ \text{VC}(\mathcal{H}) \leq K_2 M^2 d^2 N \log(dMN) < \infty.$$

We can apply Theorem 3.1 and this concludes the proof. □

References

[1] J.-Y. Audibert and A. B. Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2), apr 2007.

[2] A. R. Barron. Neural net approximation. *7th Yale Workshop on Adaptive and Learning Systems*, 1:69–72, 1992.

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

[4] A. R. Barron. Approximation and estimation bounds for artificial neural networks. *Machine learning*, 14(1):115–133, 1994.

[5] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

[6] P. L. Bartlett, V. Maiorov, and R. Meir. Almost linear VC dimension bounds for piecewise polynomial networks. *Proceedings of the 11th International Conference on Neural Information Processing Systems*, page 190–196, 1998.

[7] P. L. Bartlett and M. Wegkamp. Classification with a reject option using a hinge loss. *J. Mach. Learn. Res.*, 9:1823–1840, jun 2008.

[8] I. Blaschzyk and I. Steinwart. Improved classification rates under refined margin conditions. *Electronic Journal of Statistics*, 12(1):793 – 823, 2018.

[9] A. Caragea, P. Peteren, and F. Voigtlaender. Neural network approximation and estimation of classifiers with classification boundary in a Barron class. *accepted for publication in Annals of Applied Probability*, 2022.

[10] W. E and S. Wojtowytsch. Representation formulas and pointwise properties for Barron functions. *Calculus of Variations and Partial Differential Equations*, 2022.

[11] S. Gey. Risk bounds for CART classifiers under a margin condition. *Pattern Recognition*, 45(9):3523–3534, sep 2012.

[12] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[13] Y. Kim, I. Ohn, and D. Kim. Fast convergence rates of deep neural networks for classification. *Neural Networks*, 138:179–197, 2021.

[14] Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[15] E. Massart and E. Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5):2326–2366, 2009.

[16] P. Petersen and F. Voigtlaender. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, 108:296–330, 2018.

[17] M. Qian and S. Murphy. Performance guarantees for individualized treatment rules. *The Annals of Statistics*, 39(2):1180–1210, 2011.

[18] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning*. Cambridge University Press, 2014.

[19] S. Smale and D.-X. Zhou. Learning theory estimates via integral operators and their approximations. *Constructive Approximation*, 26:153–172, 08 2007.

[20] I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

[21] I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6(8):211–232, 2005.

[22] A.B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

[23] A.B. Tsybakov and E. Mammen. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.

[24] V. Vapnik. Pattern recognition using generalized portrait method. *Automation and Remote Control*, pages 774–780, 1963.

[25] R. Werpachowski, A. György, and C. Szepesvári. Detecting overfitting via adversarial examples. *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

[26] S. Zheng, P. Wu, A. Goswami, M. Goswami, D. Metaxas, and C. Chen. Error-bounded correction of noisy labels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11447–11457. PMLR, 13–18 Jul 2020.

A Proof of Proposition 3.2

This proof is a reformulation of the proof of [[9], Proposition 2.2], itself a modified version of the proof of [[2], Theorem 2]. The main difference with [9] lays in the order of the arguments and in the notation.

Proof. The main idea of the proof is to rewrite a Barron function as an expectation of indicators of half-spaces. By definition of what a Barron function is, this expectation is bounded and indicators of half-spaces can be well approximated by ReLU functions, that is by neural networks. Therefore one can bound the norm of the difference between the actual function and its approximation by ReLU neural networks. There are three steps : first writing f as an expectation. Then approximating this expectation by expectation of neural networks and derive the bound from this approximation.

First step - writing f as an expectation of half-spaces : Let us define $f_0 : X_0 \to \mathbb{R},\ f_0(x) = f(x+x_0) - c$ and $F_0 : \mathbb{R}^d \to, F_0(\xi) = e^{i\langle x_0, \xi\rangle} F(\xi)$. We also note $\Omega = (\mathbb{R}^d \setminus \{0\}) \times [0,1]$ and $\xi^* = \xi/|\xi|_{X_0}$. Using the polar representation of the Fourier transform $F_0(\xi) = |F_0(\xi)|e^{i\theta_\xi}$, applying a change of variable and taking the real part of the obtained integral, one can rewrite f_0 as :

$$f_0(x) = \int_\Omega \left(\mathbb{1}_{(0,\infty)}(-\langle \xi^*, x\rangle - t) - \mathbb{1}_{(0,\infty)}(\langle \xi^*, x\rangle - t)\right) \cdot |F_0(\xi)| \cdot \sin(t|\xi|_{X_0} + \theta_\xi) \cdot |\xi|_{X_0} dt d\xi.$$

Defining :

- $s(\xi, t) = \operatorname{sign}(\sin(t|\xi|_{X_0} + \theta_\xi))$,
- $\gamma(\xi, t) = |F_0(\xi)| \cdot \sin(t|\xi|_{X_0} + \theta_\xi) \cdot |\xi|_{X_0}$,
- $\Gamma_x(\xi, t) = \mathbb{1}_{(0,\infty)}(-\langle \xi^*, x\rangle - t) - \mathbb{1}_{(0,\infty)}(\langle \xi^*, x\rangle - t)$,
- $||u(\xi, t)||_\Omega = \int_\Omega |u(\xi, t)| d\xi dt$ for a measurable function u,

we can rewrite f_0 again :

$$f_0(x) = ||\gamma(\xi, t)||_\Omega \cdot \int_\Omega \Gamma_x(\xi, t) \cdot s(\xi, t) \cdot \frac{|\gamma(\xi, t)|}{||\gamma(\xi, t)||_\Omega} dt d\xi.$$

Note that $p(\xi, t) := \frac{|\gamma(\xi,t)|}{||\gamma(\xi,t)||_\Omega}$ is a probability density function, well-defined since $||\gamma(\xi, t)||_\Omega > 0$. We derive from this the probability measures

$$d\mu_\pm := \frac{\mathbb{1}_{s(\xi,t)=\pm 1} \cdot p(\xi, t)}{||\mathbb{1}_{s(\xi,t)=\pm 1} \cdot p(\xi, t)||_\Omega} dt d\xi$$

and the following functions $f_\pm(x) := \int_\Omega \Gamma_x(\xi, t) d\mu_\pm(\xi, t)$ such that :

$$f_0 = v \cdot (V_+ \cdot f_+ - V_- \cdot f_-), \tag{12}$$

where $v = ||\gamma(\xi, t)||_\Omega$ and $V_\pm = ||\mathbb{1}_{s(\xi,t)=\pm 1} \cdot p(\xi, t)||_\Omega$.

We now have an expression of f_0 as linear combination of expectations of half-spaces. Note that bounding the approximation by a single-layered neural network with $4N$ neurons of the $f_\pm$ suffices to reach the general approximation, since this linear combination is easily realizable with a neural network. Let us therefore prove that for some ${}_0 \in \mathbb{R}$: $||f_\pm - R_\sigma \Phi_\pm||_{\sup} \leq N^{-1/2} \cdot \left(\frac{C}{vV_\pm} + \kappa_0\sqrt{d}\right)$, with $4N$ neurons and weights bounded by $4 + \vartheta(X, x_0)$.

Second step - approximating f by an expectation of ReLU neural networks. We approximate the indicators of half-spaces by some neural networks, which allows us to approximate f, expectation of indicators of half-spaces as seen in 12, by an expectation of neural networks. First, let us define for $\varepsilon > 0$:

$$H_\varepsilon : \mathbb{R} \to [0,1],\ H_\varepsilon(x) = \frac{1}{\varepsilon}\left(\sigma(x) - \sigma(x - \varepsilon)\right),$$

and remark that $H_\varepsilon = \mathbb{1}_{(0,\infty)}$ on $\mathbb{R} \setminus (0, \varepsilon)$. We can use it to approximate Γ_x with :

$$N_{\varepsilon,x} : \Omega \to [-1,1],\ N_{\varepsilon,x}(\xi, t) = H_\varepsilon(-\langle \xi^*, x\rangle - t) - H_\varepsilon(\langle \xi^*, x\rangle - t).$$

Note that $N_{\varepsilon,x} = \Gamma_x$ on Ω with $t \notin J^{(\varepsilon)}_{\xi,x} := [-\langle \xi^*, x\rangle - \varepsilon,\ -\langle \xi^*, x\rangle] \cup [\langle \xi^*, x\rangle - \varepsilon,\ \langle \xi^*, x\rangle]$. Using the following bounds $0 \leq p(\xi,t) \leq v^{-1}|\xi|_{X_0}|F(\xi)|$ and (by definition of the Barron class) $\int_{\mathbb{R}^d} |\xi|_{X_0}|F(\xi)|d\xi \leq C$, we have :

$$\begin{aligned}\Big| f_\pm(x) - \int_\Omega N_{\varepsilon,x}(\xi,t)\, d\mu_\pm(\xi,t)\Big| &\leq \int_{\mathbb{R}^d\setminus\{0\}} \int_0^1 2\cdot \mathbb{1}_{J^{(\varepsilon)}_{\xi,x}}(t)\cdot \frac{1}{V_\pm}\, p(\xi,t)\, dt\, d\xi \\ &\leq \frac{4\varepsilon}{v\, V_\pm} \int_{\mathbb{R}^d} |\xi|_{X_0}\cdot |F(\xi)|\, d\xi \leq \frac{4\varepsilon C}{v\, V_\pm}.\end{aligned}$$

Choosing $\varepsilon := \frac{1}{4}N^{-1/2}$ and defining $f_{\pm,\varepsilon} : X_0 \to \mathbb{R},\ f_{\pm,\varepsilon}(x) = \int_\Omega N_{\varepsilon,x}(\xi,t)\, d\mu_\pm(\xi,t)$, the above inequality yields $\|f_\pm - f_{\pm,\varepsilon}\|_{\sup} \leq N^{-1/2}\cdot \frac{C}{v\, V_\pm}$.

Third step - Using bounds for empirical processes to complete the proof. Let us denote by $\mathcal{N}$ the class of shallow neural networks with four neurons on the hidden layer such that for $\Phi \in \mathcal{N}, R_\sigma\Phi(x) = (\xi,t) \to N_{\varepsilon,x}(\xi,t) - \lambda,\ \lambda \in \mathbb{R}$. There exists a bound on the VC-dimension of this class [[5], Theorem 6] :

$$\mathrm{VC}(\{\mathbb{1}_{g>0} : g \in \mathcal{N}\}) \leq \kappa_1 d.$$

Note that the fact that we are actually using $\xi^* = \frac{\xi}{|\xi|_{X_0}}$ implies the use of a map $\Theta : \Omega \to \mathbb{R}^d \times [0,1], \Theta(\xi,t) = (\xi^*,t)$ to have for all $\lambda \in \mathbb{R}$:

$$\{\mathbb{1}_{N_{\varepsilon,x}>\lambda} : x \in X_0\} \subset \{\mathbb{1}_{g\circ\Theta>0} : g \in \mathcal{N}\}.$$

This map does not change the VC dimension and thus : $\mathrm{VC}(\{\mathbb{1}_{N_{\varepsilon,x}>\lambda} : x \in X_0\}) \leq \kappa_1 d$.

Now, applying the following proposition [[9], Proposition A.1]:

Proposition A.1. *There is a universal constant $\kappa > 0$ with the following property: If $(\Omega, \mathcal{F}, \mu)$ is a probability space, if $a, b \in \mathbb{R}$ with $a < b$, and if $\emptyset \neq \mathcal{G} \subset \{g : \Omega \to [a,b] \colon g \text{ measurable}\}$ satisfies*

$$d := \sup_{\lambda\in\mathbb{R}}(\{I_{g,\lambda}\colon g \in \mathcal{G}\}) < \infty, \qquad \textit{where} \qquad I_{g,\lambda} : \quad \Omega \to \{0,1\}, \quad \omega \to \mathbb{1}_{g(\omega)>\lambda},$$

then for any $n \in \mathbb{N}$ and $S = (X_1, \ldots, X_n) \overset{i.i.d.}{\sim} \mu$, we have

$$\mathbb{E}_S\Big[\sup_{g\in\mathcal{G}}\Big|\mathbb{E}_{X\sim\mu}[\,g(X)\,] - \frac{1}{n}\sum_{i=1}^n g(X_i)\Big|\Big] \leq \kappa\cdot(b-a)\cdot\sqrt{\frac{d}{n}},$$

we can derive the following bound, since $\mathbb{E}_{(\xi,t)\sim\mu_\pm}[N_{\varepsilon,x}(\xi,t)] = f_{\pm,\varepsilon}(x)$:

$$\mathbb{E}\Big[\sup_{x\in X_0}\big|f_{\pm,\varepsilon}(x) - N^{-1}\textstyle\sum_{i=1}^N N_{\varepsilon,x}(\xi_i,t_i)\big|\Big] \leq \kappa_2\cdot\sqrt{\frac{\kappa_1 d}{N}}.$$

for $(\xi_i, t_i) \overset{\text{i.i.d}}{\sim} \mu_\pm$.

Using that for a random variable defined on a probability space Ω with a distribution P, $\mathbb{E}_P(X(\omega)) \leq a$ implies that there exists $\hat{\omega} \in \Omega$ such that $X(\hat{\omega}) \leq a$, we have for one specific realization $\big((\xi_1,t_1),\ldots,(\xi_N,t_N)\big) \in \Omega^N$ that

$$\sup_{x\in X_0}\Big|f_{\pm,\varepsilon}(x) - \frac{1}{N}\sum_{i=1}^N N_{\varepsilon,x}(\xi_i,t_i)\Big| \leq \kappa\sqrt{d}\,N^{-1/2}.$$

Since we have :

$$\frac{1}{N}N_{\varepsilon,x}(\xi_i,t_i) = \frac{\varepsilon^{-1}}{N}\cdot\Big(\sigma\big(-\langle\xi_i^*,x\rangle - t_i\big) - \sigma\big(-\langle\xi_i^*,x\rangle - t_i - \varepsilon\big) - \sigma\big(\langle\xi_i^*,x\rangle - t_i\big) + \sigma\big(\langle\xi_i^*,x\rangle - t_i - \varepsilon\big)\Big),$$

we can conclude that the average $\frac{1}{N}\Sigma_{i=1}^N N_{\varepsilon,x}(\xi_i,t_i)$ that approximates well $f_\pm$ can be implemented by a shallow ReLU network with $4N$ neurons in the hidden layer. Setting $R_\sigma\Phi_\pm(x) := \frac{1}{N}\Sigma_{i=1}^N N_{\varepsilon,x}(\xi_i,t_i)$ and defining :

$$R_\sigma\Phi(x) := c + v\,V_+\cdot R_\sigma\Phi_+(x-x_0) - v\,V_-\cdot R_\sigma\Phi_-(x-x_0).$$

Because of $f(x) = c + f_0(x - x_0) = c + v\,V_+ \cdot f_+(x - x_0) - v\,V_- \cdot f_-(x - x_0)$ and $0 < v \le C$, we have :

$$\begin{aligned}\|f - R_\sigma \Phi\|_{\sup} &\le N^{-1/2} \cdot \Big(v\,V_+ \cdot \big(\tfrac{C}{v\,V_+} + \kappa_0\sqrt{d}\big) + v\,V_- \cdot \big(\tfrac{C}{v\,V_-} + \kappa_0\sqrt{d}\big)\Big) \\ &= N^{-1/2} \cdot \big(2C + v\kappa\,\sqrt{d}\big) \le \big(2 + \kappa_0\,\sqrt{d}\big) \cdot C \cdot N^{-1/2} \le \kappa\,\sqrt{d} \cdot C \cdot N^{-1/2}\end{aligned}$$

Regarding the weights, recalling that by definition $||\xi_i^*||_\infty \le \vartheta(X, x_0)$, $|t_i| \le 1$ and $\varepsilon = \frac{1}{4}N^{-1/2}$ implying that $\varepsilon^-1/N \le 4$, we can see that the weights are bounded by $4 + \vartheta(X, x_0)$. Note that the change of variable $x - x_0 \to x$ increases the bound for the magnitude of the weights by a $(1 + ||x_0||_1)$ factor.

□

B Neural network architecture and bounding the weights in 3.7

In this appendix we give the details of the architecture and weights boundedness of the neural network I_N in Theorem 3.7. Our network being almost the exact same as in [[9], Theorem 3.7], our proof is an adaptation of the one presented in this paper.

We can use the Figure 5 to visualize the neural network approximating $\mathbb{1}_\Omega$ on a rectangle Q_m.

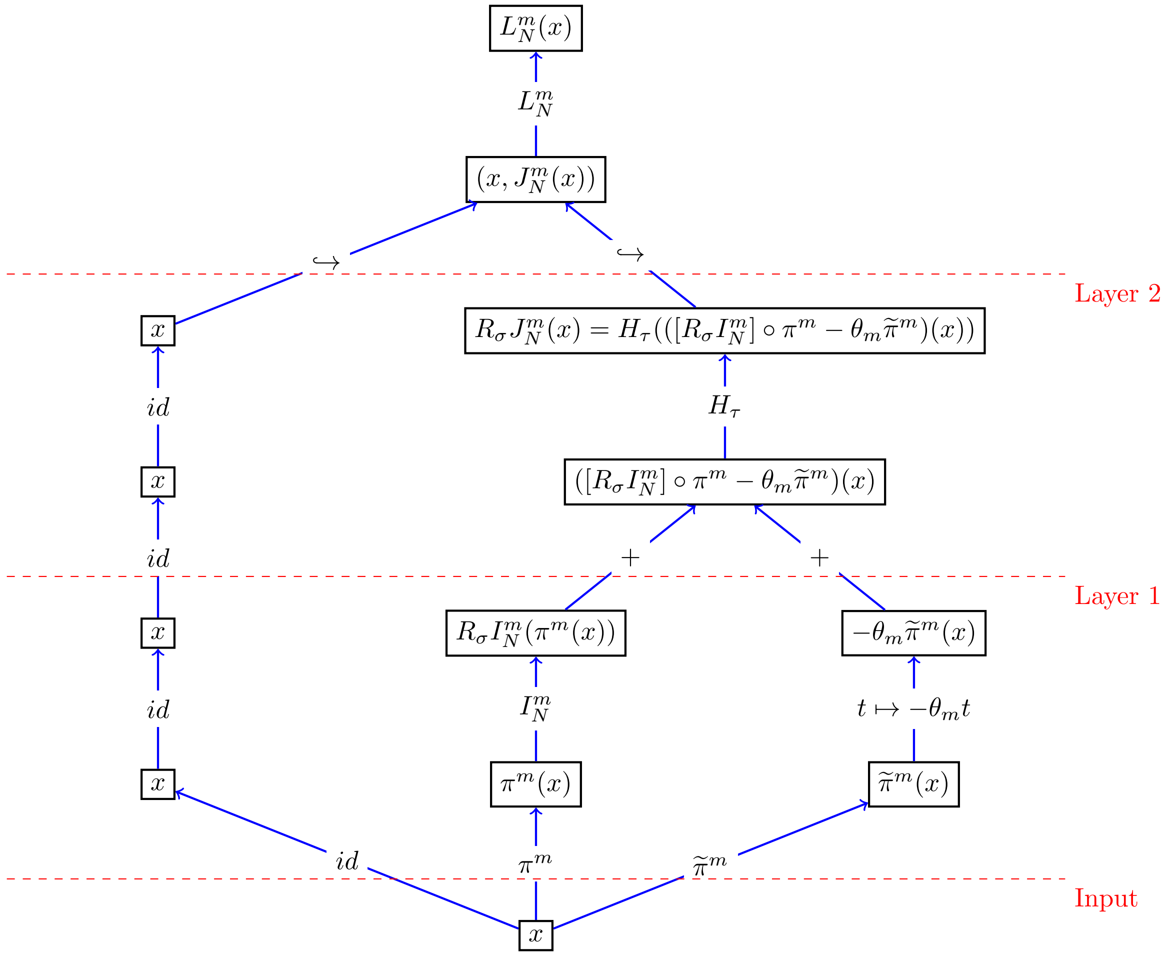

Figure 5: Visualization of the neural network L_N^m for the case of a rectangle Q_m. This figure is taken from the proof of [[9] Theorem 3.7]

In the following, we explicitly describe each of the layers of the network computing L_N^m; we then describe how these networks are combined to obtain I_N.

Description of the layers :

- The input layer will be of dimension d since the input x is in $\mathbb{R}^d$.

- The first hidden layer consists of $2d$ neurons computing $\sigma(\pm x_i), i \in 1, d$, N ones computing $R_{{}^{m}_{N}(x^{i})}$ and 2 computing $\sigma(\pm\theta_m x_i)$, respectively. Note that pi^m and $\hat{\pi}^m$ are the projections of x to x^i and x_i.
- The second hidden layer will compute $R_\sigma J_N^m$, as well as the $t_i(u_i)$ to compute in the next layer $R_\sigma L_N^m$. The latters need $4d$ neurons (four per coordinate) to be computed and $R_\sigma J_N^m(x) = H_\tau\big(R_\sigma I_N^m(\pi^m(x)) - \theta_m \widetilde{\pi}^m(x)\big)$ can be computed with two neurons in the following way : $R_\sigma J_N^m(x) = \frac{1}{\tau}(\psi_1(x) - \psi_2(x))$ where

$$\psi_1(x) := \sigma\Big(D + \sum_{k=1}^{N} C_k \phi_k(x) - \sigma\big(\theta_m \widetilde{\pi}^m(x)\big) + \sigma\big(-\theta_m \widetilde{\pi}^m(x)\big)\Big)$$

  and

$$\psi_2(x) := \sigma\Big(D + \sum_{k=1}^{N} C_k \phi_k(x) - \sigma\big(\theta_m \widetilde{\pi}^m(x)\big) + \sigma\big(-\theta_m \widetilde{\pi}^m(x)\big) - \tau\Big),$$

  and $\tau = \gamma N^{-1/2}$.
- The third hidden layer is made of one neuron computing :

$$\eta_\varepsilon(x, R_\sigma J_N^m(x)) = R_\sigma L_N^m(x) = \sigma\bigg(\tfrac{1}{\varepsilon}\sum_{i=1}^{d}\big(t_i^1 - t_i^2 - t_i^3 + t_i^4\big)(x_i) + \tfrac{1}{\tau}\big(\psi_1(x) - \psi_2(x)\big) - d\bigg).$$

- The fourth one with one neuron is the one where the outputs of each $R_\sigma L_N^m$ are summed and smoothed with σ.
- The last layer is the output one, one neuron too.

Thus, I_N can be realized by a ReLU neural network with 4 hidden layers, architecture

$$\mathcal{A} = \big(d,\ M(N + 2d + 2),\ M(4d + 2),\ M,\ 1, 1\big),$$

and $d + 1 + M(N + 6d + 5) + 1 \leq 7M(N + d)$ neurons.

To estimate the number of non-zero weights of $I_N, W(I_N)$, a bound can be found by taking the product of the number of neurons on every pair of consecutive layers in the L_N^m networks, summing up over the layers, multiplying by M, adding M to account for the weights of the final output layer, and finally adding the total number of non-input neurons to account for the biases. This yields

$$W(I_N) \leq M \cdot \big(d(N + 2d + 2) + (N + 2d + 2)(4d + 2) + (4d + 2) \cdot 1\big) + M + MN + 6Md + 5M + 1,$$

which gives the following estimation : $W(I_N) \leq 54Md^2N$.

Bounding the magnitude of the weights and biases:

We use for this the definition of ϑ in Proposition 3.2, namely $\vartheta(Q_m, q_m) \leq \gamma^{-1}N^{1/2}$. Moreover $\|q_m\|_1 \leq (d-1)R$, so the magnitudes of weights and biases for I_N^m, are bounded as follows :

$$\sqrt{B} \cdot (6 + \vartheta(Q_m, q_m) + \|q_m\|_1) \leq \sqrt{B} \cdot (6 + \gamma^{-1}N^{1/2} + dR),$$

and therefore the overall bound for the first layer is $\sqrt{B} \cdot (6 + \gamma^{-1}N^{1/2} + dR) + 1$.

For the second layer, the weights corresponding to the first $4d$ neurons are bounded by $1 + \varepsilon + R$ and for the last 2 neurons again by $1 + \sqrt{B} \cdot (6 + \gamma^{-1}N^{1/2} + dR)$. Finally for the third layer, the weights and biases are bounded by $\max(\frac{1}{\varepsilon}, \frac{1}{\tau}, d) \leq d + \gamma^{-1}N^{1/2}$.

Wrapping everything up and using classical estimates such $\sqrt{B} \leq 1 + B$ and the fact that $d \geq 2$, the weights of I_N have magnitudes bounded by

$$\begin{aligned}&\max\big\{1 + 6\sqrt{B} + \sqrt{B}\gamma^{-1}N^{1/2} + \sqrt{B}dR,\ 1 + \varepsilon + R,\ d + \gamma^{-1}N^{1/2}\big\}\\ &\leq d(4 + R)(1 + B) + \sqrt{N} \cdot \big(B^{-1} + B^{-1/2}\big).\end{aligned}$$