Butlletí de la Societat Catalana de Matemàtiques
Vol. 36, núm. 1, 2021. Pàg. 5–67. DOI: 10.2436/20.2002.01.97

Algorithmic Learning and Deep Neural Networks

Joan Bruna* Sebastià Xambó-Descamps**

2021 (Catalan version), April 2026 (English translation with the help of Gemini)

Abstract

In this article, you will find a description of the nature of algorithmic learning, as well as its most relevant modalities, and a presentation of the main mathematical ingredients that serve as its foundation, both for the definition and study of models and for the analysis of algorithms. You will also find an extensive bibliography and some recommendations for further study.

Keywords: algorithmic learning, curse of dimensionality, neural networks, stochastic gradient descent, VC dimension, Rademacher complexity, double descent, causality and explainability, hypercomplex networks.

MSC2010 Classification: 68T01, 68Q32, 68T05, 62F15, 68W25, 82C32, 62M45, 65K10, 90C26, 93E35, 30G35.

Contents

Introduction

In this section we explain, in a somewhat informal language, some of the basic ideas of *algorithmic learning* (AL), or *machine learning* (ML), terms to which we attribute, as a first approximation, the same meaning; that is to say, the search for efficient algorithms that return, from a set of data (experience), accurate predictors of information associated with new data.

In usual terms, AL forms part of the field of so-called *artificial intelligence* (AI) and comprises the study of *neural networks* (NN). For general references on these matters, see Appendix B (page 43). This appendix contains a subsection for each section of the article, starting with §B.0, where you can find a list of treatises on the issues that will occupy us from now on. In each of these subsections you can find, beyond the citations inserted in the corresponding section of this article, complementary references, bibliographic comments and, where appropriate, brief definitions, at the beginning of the subsection, of concepts that may be considered somewhat specialized.

There are several types of AL. In this section we will only consider the case of *supervised* AL. The purpose of this modality is the creation and investigation of algorithms that return a function $f : X \to Y$ with an acceptable capacity to predict the values of a function $f^* : X \to Y$, unknown to the algorithm and which we will call *expert* or *supervisor*, based on a series of examples, that is to say, a finite set of pairs

$$\mathcal{D} = \{(x^j, y^j) : x^j \in X, y^j = f^*(x^j) \in Y, j \in [n]\}, \quad [n] = 1, \dots, n.$$

We will often refer to such an algorithm as the *learner*, since its task is to achieve an extrapolation of the examples that reasonably predicts the values produced by the expert. It is important to note that for the treatment of uncertainties regarding the y^j it is necessary to resort to the more general model of which we give notice in subsection B.2.

There are two types of supervised AL problems that have fundamental relevance: *classification*, when Y is finite (a set of *classes*), and *interpolation* or *regression*, when $Y = \mathbb{R}$. Formally, we can treat both cases by setting $f(x) \simeq f^*(x)$ to indicate $f(x) = f^*(x)$ in the case of classification and $f(x) \approx f^*(x)$ in the case of regression, where $\approx$ denotes approximate equality according to a prefixed criterion. With this convention, we can measure the goodness of f with the *learning rate*, that is, the proportion a of cases in which $f(x^j) \simeq f^*(x^j) = y^j$ $(j = 1, \dots, n)$, and with the *accuracy*, which we can also call the *generalization rate* or *predictive capacity*, that is, the proportion g of cases in which $f(x) \simeq f^*(x)$ $(x \in X)$. Thus, $1 - a$ is the learning error rate and $1-g$, the generalization error rate. In practice, g is substituted by the proportion of cases in which $f(\bar{x}^j) \simeq \bar{y}^j$, where $\overline{\mathcal{D}} = \{(\bar{x}^j, \bar{y}^j = f^*(\bar{x}^j)) : \bar{x}^j \in X, j \in [\bar{n}]\}$ is a set of examples (*test examples*) generated independently of $\mathcal{D}$, and one of our purposes is to analyze the conditions of validity of this procedure.

Let us note that achieving a learning rate $a = 1$ is equivalent to a perfect memorization of the examples $\mathcal{D}$, a fact that entails, in the most well-known

models, a poor behavior in the face of new examples, which in practice means the examples $\overline{\mathcal{D}}$. Therefore, we must expect that the best use of the examples $\mathcal{D}$ can often only be produced with an appropriate value of the learning rate. Beyond this value, *overfitting* appears and, below it, *underfitting*. Finding this optimal value of the learning rate is one of the merits that a ML algorithm must have. These concepts are illustrated and commented on in Figure 1.

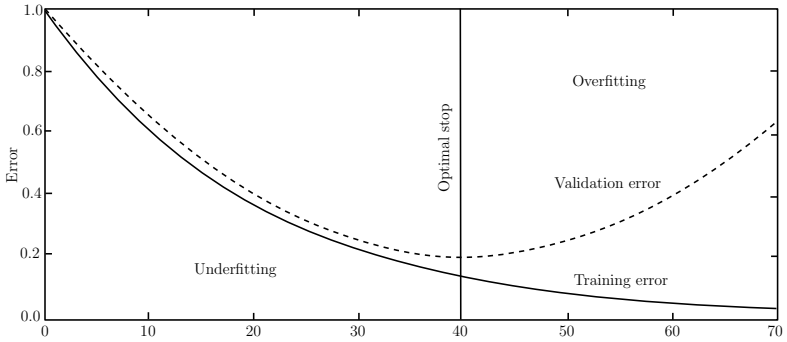

Figure 1: We are assuming that learning, such as in a linear regression, takes place in a loop, the cycles of which are called *epochs* (abscissa axis), and that in each cycle the learning rate on the data increases. At the beginning, the validation error also decreases, which is how we estimate the generalization capacity, but a moment arrives when this error increases again. This moment is that of the optimal stop. Before, one is still in the regime of underfitting and after, in that of overfitting, in which the cycles increase the memorization of the training data, including all irrelevant aspects (such as the presence of some type of "noise"), to the detriment of the generalization capacity. This scheme will be modified in §5 to include some subtleties of the generalization problem discovered in recent years (Figure 9).

It is a fact that techniques are known for creating algorithms with good learning and generalization rates, at least for certain problems. Currently, this fact is news almost every day and in various domains, and is especially notorious, due to the media coverage, the accomplishments of champion algorithms in games and competitions of all kinds. Among these, the cases of Go and poker have recently drawn attention and, more recently, the case of algorithms that win at console games. But perhaps even more spectacular are the results in the case of images (biomedical predictors, recognition of people, autonomous vehicles) and of languages (speech or writing), especially for the implications they have for social and economic dynamics everywhere.

Example: Regularized linear regression. Suppose that the data x are vectors and that the values y are real numbers. Let w be a (unknown) vector of *weights* (one weight for each of the n components of the vectors x), and we set $f^* = f_w$, where

$$f_w(x) = w \cdot x = w_1 x_1 + \cdots + w_n x_n$$

(a *weighted sum* of the values of x). In order to optimize the learning rate, we can take as vector w that which is provided by the method of *least squares*:

$$\operatorname{argmin}_w \sum_{i=1}^{n} (w \cdot x^i - y^i)^2. \tag{1}$$

Note that the minimum is zero if, and only if, $w \cdot x^i = y^i$ for all i, which is equivalent to saying that the learning rate is 1. To keep overfitting at bay, and thus improve predictive capacity, one can resort to a *regularized* version of (1), which means choosing w according to the relation

$$\operatorname{argmin}_w \sum_{i} (w \cdot x^i - y^i)^2 + \lambda \|w\|_2^2, \tag{2}$$

where λ is a positive real number that in practice is determined by the learning algorithm. Let us remark that a small λ favors overfitting, while a large λ entails that $|w|$ has to be small and, therefore, w has fewer possibilities of achieving a good learning rate. The solution of expressions (1) and (2) can be obtained by standard optimization algorithms (see §4).

Examples of classification. Perhaps one of the most familiar is the problem of filtering email messages according to whether they are *spam* or not. Another example is the recognition of handwritten characters, in which predictive capacity has the flexibility to adapt to an infinity of possible variations. Of special interest for mathematicians, we have programs that transform the image of a mathematical text into a LaTeX version of it (see `https://mathpix.com/`), or mobile applications that give information about objects using their photograph, or that transform speech in one language into written text and can also translate it, both voice and writing, to another language.

Logistic regression. We say that the probability $p = p(x)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, follows a *logistic regression* if the log of the *odds* of x, $p/(1-p)$, follows a linear regression:

$$\log \frac{p}{1-p} = w_1 x_1 + \cdots + w_d x_d = w \cdot x.$$

Once the weights w of this linear regression are found (normally, using tables of previous results in which the a list of feature vectors x are paired with their frequencies), we then obtain

$$p = \frac{1}{1 + e^{-w \cdot x}},$$

a relationship that can serve to make predictions about the probability of any new feature vector x. The function $\sigma(t) = 1/(1 + e^{-t})$ has a sigmoid shape and is also known by the name of *logistic function* (Figure 2). Sometimes it is convenient to use the function $2k\sigma(t) - k = k(1 - e^{-t})(1 + e^{-t})^{-1}$, which has a sigmoid shape taking values in $(-k, k)$, and which we can denote by σ_k. For example, $\sigma_{1/2} = \sigma - 1/2$ varies between $-1/2$ and $1/2$, while σ_1 varies between -1 and 1.

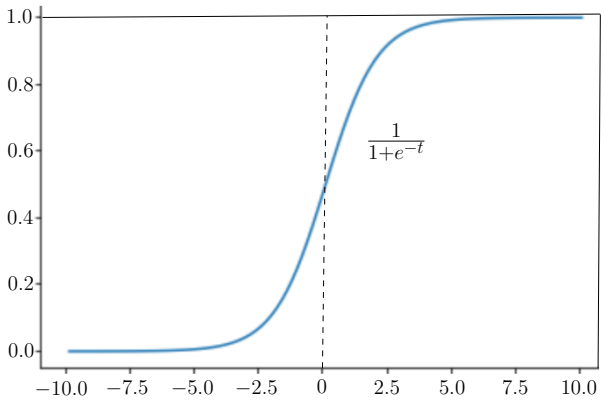

Figure 2: Graph of the logistic function.

Organization of the content. In the first section, some ideas of a relatively elementary nature are gathered, with a long tradition in the study of mathematics and statistics, which can be seen as the roots of mathematical learning models and which in any case continue to play a role in the most current theories.

The basic model on which we base the rest of the article is presented in §2. The main challenge of AL in the face of the massive amounts of available data is the so-called *curse of dimensionality*, which we briefly consider in the first subsection, §2.1. The ingredients of the basic model and their significance are described in subsection §2.2, which ends with what we call the *fundamental theorem* of AL. In the last two subsections, mathematical models of neurons and neural networks and their standing as universal approximators are described.

The state of the art of the theoretical foundations of AL is broken down in the following three sections: §3 (Approximation), §4 (Optimization) and §5 (Generalization). Given the importance of these topics, we defer to the respective introductions the description of their scope in more detail.

In the last section, §6, we refer to some other promising models. We also mention various open problems. In Appendix A, we collect some notions of probability used previously. Having already explained the purpose of Appendix B, it only remains for us to say that the article ends with the reference section, which contains the cited bibliography.

Finally, it should be noted that we omit practically all proofs of the results included here, but we endeavor to compensate for this with appropriate references, either within the context of each statement or in the corresponding section of Appendix B.

1 Preludes

Although the main theme of this article is Algorithmic Learning (AL) based on neural networks and the results upon which they are based, we have found

it appropriate to include this preliminary section to account for some basic notions that have played, and continue to play, an important role in the evolution of learning theories and the development of what is called *data science*. The primary motivation for this section is to act as a buffer between more or less familiar knowledge and the subsequent sections of a more technical nature. For references, see §B.1.

1.1 The Bayesian Pathway

The well-known *Bayes-Laplace rule*, often called simply *Bayes' formula*, is in fact one of the pioneering and fundamental results of learning theory.

Given events X, Y, we have:

$$P(X,Y) = \begin{cases} P(X) \cdot P(Y|X), \\ P(Y) \cdot P(X|Y), \end{cases}$$

and these relations are equivalent to:

$$\frac{P(X|Y)}{P(X)} = \frac{P(Y|X)}{P(Y)} = \frac{P(X,Y)}{P(X) \cdot P(Y)}$$

Let $L(X,Y)$ denote this value, which is symmetric in X and Y. We can then write:

$$P(Y|X) = P(Y)L(X,Y),$$

which is the *Bayes-Laplace rule*. We see that $L(X,Y)$ is the factor by which we must multiply the probability $P(Y)$ of Y, *prior* to the observation of X, to obtain the probability $P(Y|X)$ of Y *posterior* to the observation of X. In other words, $L(X,Y)$ indicates the degree of learning about Y provided by the fact of having experienced X.

Bayesian Classification: The MAP Rule. If an event X has been observed, and this fact can be attributed to one of the hypotheses $Y_1, \ldots, Y_r$, the *maximum a posteriori* rule (MAP) selects the Y_j that maximizes $P(Y_j|X)$. By the Bayes-Laplace rule, this is equivalent to choosing Y_j that maximizes $P(X|Y_j)P(Y_j)$. In the special case where the Y_j are equiprobable, the aim is to maximize $P(X|Y_j)$.

A particular case is that of a binary classification of the objects of $\mathcal{X}$. Suppose $\mathcal{X} = \mathcal{X}_0 \sqcup \mathcal{X}_1$, but that the observation of an $x \in \mathcal{X}$ does not determine with certainty (generally due to the presence of noise or other causes) to which of the two classes it belongs. The statistical model of this situation is that the attribution of a class $y \in \{0,1\}$ to an observation x is governed by a joint probability $P(x,y) = P(x|y)P(y)$. That is, we assume that the probabilities $P(0)$ and $P(1)$ are known, as well as the probabilities $P(x|y)$ that express the uncertainty regarding the class y of x. Once uncertainty is quantified this way, the optimal decision rule is the so-called *Bayes classifier*, $f^B : \mathcal{X} \to \{0,1\}$, defined by the

maximum posterior probability $P(y|x) = P(x|y)P(y)/P(x)$; that is:

$$f^B(x) = \begin{cases} 0 & \text{if } P(0|x) \geqslant P(1|x), \\ 1 & \text{if } P(0|x) < P(1|x). \end{cases}$$

This classifier is optimal for the distribution P, and its error rate is $R^* = \int \min\{P(0|x), P(1|x)\}P(x)dx$ on average. Without further information about P, one can only ensure that $0 \leqslant R^* \leqslant 1/2$.

The Bayesian Perspective of AL. Starting from data $\mathcal{D}$, a possible model $\mathcal{M}$ to explain them, and a set of parameters or weights W of the model, the Bayes-Laplace rule gives us:

$$P(W|\mathcal{D}, \mathcal{M}) = \frac{P(\mathcal{D}|W, \mathcal{M})P(W|\mathcal{M})}{P(\mathcal{D}|\mathcal{M})},$$

where $P(W|\mathcal{D}, \mathcal{M})$ is the probability of W after observing the data $\mathcal{D}$ assuming the model $\mathcal{M}$. On the right side of the equality we have $P(\mathcal{D}|W, \mathcal{M})$, the likelihood of the data $\mathcal{D}$ according to the value of the weights W relative to the model $\mathcal{M}$; $P(W|\mathcal{M})$, the *a priori* probability of the parameters W with respect to the model $\mathcal{M}$; and $P(\mathcal{D}|\mathcal{M})$, the marginal likelihood or evidence of the model, that is, $\int P(\mathcal{D}|W, \mathcal{M})P(W, \mathcal{M})dW$. The posterior probability becomes the *a priori* probability for processing the information acquired when considering new data.

A model $\mathcal{M}$ already learned can be used to predict the probability of new data $\mathcal{D}'$:

$$P(\mathcal{D}'|\mathcal{D}, \mathcal{M}) = \int P(\mathcal{D}'|W, \mathcal{D}, \mathcal{M})P(W|\mathcal{D}, \mathcal{M})dW.$$

The paradigm also allows for contrasting different models $\mathcal{M}$ in relation to the data $\mathcal{D}$:

$$P(\mathcal{M}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{M})P(\mathcal{M})}{P(\mathcal{D})} \quad \text{and} \quad P(\mathcal{D}|\mathcal{M}) = \int P(\mathcal{D}|W, \mathcal{M})P(W|\mathcal{M})dW.$$

1.2 Principal Component Analysis (PCA)

Let X be an $m \times n$ real matrix. We can think of X as the result of observing m features of n objects, so that row X^j contains the n observations $(x_1^j, \ldots, x_n^j)$ of the j-th featue ($j \in [m]$). Equivalently, column $X_k = (x_k^1, \ldots, x_k^m)^T$ ($k \in [n]$) contains the observed values of the m features of the k-th object. In any case, $X = [X_1, \ldots, X_n] = [X^1, \ldots, X^m]^T$.

Consider the mean of X^j, $\mu_j = E[X^j]$, and the covariance of X^j and X^k, $\sigma_{jk} = \mathrm{Cov}(X^j, X^k) = E[X^j \cdot X^k] - \mu_j\mu_k$. The symmetric $m \times m$ matrix $\Sigma = \mathrm{Cov}(X) = (\sigma_{jk})$ is positive semi-definite, and we call it the covariance matrix of X. Note that $\sigma_{jj} = \mathrm{Var}(X^j)$, the variance of X^j, and that $\mathrm{Var}(X^j) = \sigma_j^2$, where $\sigma_j \geqslant 0$ is the standard deviation of X^j.

Given a unit m-vector u, it follows that $\mathrm{Var}(u^T X) = u^T \Sigma u$, and that *this value is maximized precisely when u is an eigenvector u_1 of Σ with the maximum*

eigenvalue. Under these conditions, $u = u_1$ is called the *principal component* of X. Note that $u^T X$ is the vector formed by the projections of the columns of X onto u, so that these projections capture the maximum variability of X in a single direction.

The second principal component of X is the unit eigenvector u_2 corresponding to the second eigenvalue (in non-increasing order) of Σ. This vector maximizes $\mathrm{Var}(u^T X) = u^T \Sigma u$ for unit vectors u perpendicular to u_1. Continuing this process, we obtain an orthonormal basis $u_1, \ldots, u_m$ of $\mathbb{R}^m$ such that u_r maximizes $\mathrm{Var}(u_r^T X) = u_r^T \Sigma u_r$ for unit vectors perpendicular to $u_1, \ldots, u_{r-1}$. If we use U_r to denote the matrix formed by the vectors $u_1, \ldots, u_r$, the matrix $U_r^T X$ is of type $r \times n$ and incorporates the variability of X attributable to the first r eigenvalues (ordered in a non-increasing direction) Σ.

The technique of principal components is a primary example of *dimensionality reduction*, a concept to which we will return later, and it can be understood as a form of unsupervised learning. It should also be noted that $U_r^T X$ can be used as a form of preprocessing applicable to data X before subjecting them to a supervised learning procedure. The value of r is chosen such that $u_{r+1}, \ldots, u_m$ play a negligible role in explaining the variability of X.

1.3 Singular Value Decomposition (SVD)

Let X be a real matrix $m \times n$, which we think of as data in the style of what we saw in §1.2, and let r be its rank. The symmetric matrices XX^T and $X^T X$, of type $m \times m$ and $n \times n$ respectively, have rank r, are positive semidefinite and have the same nonzero eigenvalues $\lambda_1^2, \ldots, \lambda_r^2$, where we can assume that $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_r > 0$. Furthermore, if we put U and V to denote the orthonormal matrices of eigenvectors of XX^T and $X^T X$, then $X = U \Lambda V^T$, where $\Lambda_{jj} = \lambda_j$, for $j = 1, \ldots, r$, the only non-zero values of Λ. Thus, $XX^T = U(\Lambda \Lambda^T) U^T$ and $X^T X = V(\Lambda^T \Lambda) V^T$, where the first r values of the diagonal of $\Lambda \Lambda^T$ and $\Lambda^T \Lambda$ are $\lambda_1^2, \ldots, \lambda_r^2$ and all others are zero in both matrices (of type $m \times m$ and $n \times n$, respectively). Now, since $U\Lambda = (\lambda_1 u_1, \ldots, \lambda_r u_r)$, we obtain the celebrated *singular value decomposition of* X:

$$X = \lambda_1 u_1 v_1^T + \cdots + \lambda_r u_r v_r^T.$$

In fact, it turns out that for $k = 1, \ldots, r$, the matrix

$$M_k = \lambda_1 u_1 v_1^T + \cdots + \lambda_k u_k v_k^T$$

is the optimal approximation of X with matrices of rank k (Eckart-Young theorem). This result is, therefore, also a form of dimensionality reduction. The value of r is chosen so that $u_{r+1}, \ldots, u_m$ play a negligible role in explaining the variability of X in the context where it is used.

An important application of singular decomposition is that the least-squares solution to the linear system $Xa = b$ is $a = X^\dagger b$, where $X^\dagger$ is the *Moore-Penrose pseudoinverse* of X; i.e., $X^\dagger = V \Lambda^\dagger U^T$, with $\Lambda_{jj}^\dagger = \lambda_j^{-1}$ for $j = 1, \ldots, r$ and all other entries zero.

1.4 Learning by Non-Parametric Methods

While the main focus of this article is supervised learning, it is of interest to briefly present the idea of unsupervised learning. We do this by describing two algorithms: k-Means, a data clustering procedure, and k-NN (nearest neighbors), a classification procedure in which the available data are classified by an expert and the decision is achieved by looking at the k available data points closest to the data to be classified.

k-Means. In general terms, this algorithm divides a set of unlabeled n-dimensional vectors $\mathcal{D}$ into k groups by minimizing a certain cost function, but the essence of its operation is reflected in the following steps: (1) Select k different vectors $z_1, \dots, z_k$ (in principle, they can be from the data set itself). (2) Assign each vector x^j from $\mathcal{D}$ to the closest z_i; i.e., the first satisfying $d(x^j, z_i) = \min(d(x^j, z_1), \dots, d(x^j, z_k))$, resulting in an initial classification of $\mathcal{D}$ into k groups. (3) Update each z_i by taking the barycenter of the group assigned to z_i. (4) Iterate until the z_i are stable (according to a pre-set tolerance).

k-NN. Suppose we have a data set $\mathcal{D} = \{(x^j, y^j)\}$ ($j \in [n]$) where the x^j are N-dimensional vectors and y^j are elements of a finite set Y. Let k be a positive integer. Given an arbitrary vector x in the space of the x^j, the k-NN algorithm classifies it by assigning it a label as follows: (1) search for the k vectors $x^{j_1}, \dots, x^{j_k}$ such that the distances $d(x, x^{j_1}), \dots, d(x, x^{j_k})$ are the smallest possible, and (2) assign to x the *mode* of the set $\{y^{j_1}, \dots, y^{j_k}\}$. To find the set $\{x^{j_1}, \dots, x^{j_k}\}$, it suffices to make a list of pairs $(j, d^j = d(x, x^j))$, sort the d^j in non-decreasing order, and retain the first k pairs. If there is a tie in step (2), the tie is broken by choosing the one with the minimum distance. In binary classification, it is advisable to choose an odd k.

The k-NN algorithm can be easily modified so that k grows with n. The following theorem provides sufficient conditions for this modified algorithm to be a universally consistent classifier.

Theorem 1 ([187, 207])**.** *Let f_n be the binary classifier constructed with a sample of n points. If $n \to \infty$ and $k \to \infty$ so that $k/n \to 0$* (e.g., if $k = \log n$)*, then $R(f_n) \to R^*$ for any probability distribution P, where $R(f_n)$ and R^* are the error rates of f_n and of the Bayes classifier, respectively.*

However, this theorem does not have the practical consequences that might seem at a first glance, as it hides a case of the curse of dimensionality, in the sense that to achieve $R(f_n) - R^* \leqslant \epsilon$, one needs, in the worst case, a number of examples $n = O(\epsilon^{-d})$ [176]. We will see more details of this curse of dimensionality in §2.1.

2 Inductive Learning in High Dimensions

The problem of inductive learning, as we have considered it in the preceding sections, essentially reduces to a data interpolation problem. It is, therefore, a

classic problem with a long history in statistics and signal processing. What, then, makes the mathematics of AL special?

2.1 The Curse of Dimensionality

The key is the high dimension of the data spaces. Signals such as images, audio, and video live in spaces of millions of dimensions, and classic interpolation methods become useless in this regime because, to obtain consistent solutions under weak hypotheses of local regularity (such as Lipschitz), it is inevitable that the number of examples n must grow exponentially with the dimension—a phenomenon colloquially known as the *curse of dimensionality*.

Indeed, dividing the unit cube of a d-dimensional space into cubes of side ϵ results in ϵ^{-d} cubes, a quantity that grows exponentially with the dimension. A related observation is that the volume of the sphere of radius 1 in dimension d decreases very rapidly starting from $d = 5$ (see Figure 3), so familiar properties valid in the plane, in space, or in low dimensions do not hold in high dimensions.

Simply put, the curse of dimensionality lies in the fact that interpolating points in high dimensions is intrinsically more difficult than in low dimensions.

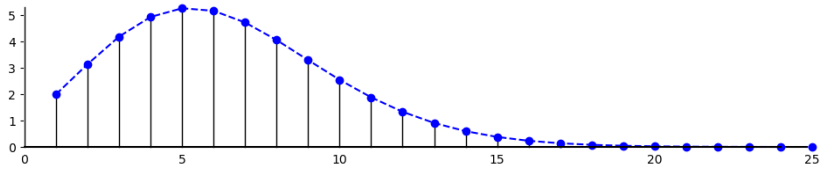

Figure 3: Volume v_n of the sphere S^n of radius 1 for $n = 1, 2, \ldots, 25$. We have $v_1 = 2$, $v_2 = \pi$ and, for $n > 2$, $v_n = \frac{2\pi}{n} v_{n-2}$. Therefore v_n increases when $n < 2\pi$ and monotonously decreases for $n > 2\pi$. In fact, v_n converges very quickly to 0 when $n \to \infty$. For example, $v_{100} \simeq 2.368 \times 10^{-40}$.

To face the curse of dimensionality, it is necessary to construct mathematical theories adapted to the treatment of high-dimensional objects. Unlike the linear and logistic regression examples from previous sections, it is first necessary to introduce non-linear approximation spaces. This poses new algorithmic challenges for defining optimal estimators in these spaces. Among the main difficulties are the inevitable presence of non-convex functions and the need to use advanced probability tools (such as measure concentration techniques) that guarantee learning.

The aim of this section is to first present, §2.2, the basic supervised AL model that we will use from now on as a backdrop and scenario for all the considerations that follow. Then, §2.3, the neuron and neural network models are introduced, which in particular allows us to have a large abundance of hypothesis spaces (one of the key elements of the basic model), and to guarantee the universal approximation property, §2.4.

We have systematized the more detailed study of the role of the basic model in overcoming the obstacles of the dimensional curse in the sections, §3, §4 and

§5, dedicated to the approximation, optimization and generalization problems, respectively.

2.2 Basic Model

Below we describe the ingredients that go into the basic mathematical model of inductive learning. As a first example, you can consider the linear regression explained in the Introduction, page 3.

Data Domain: $\mathcal{X}$, the space (or set) from which data are extracted. Information theory allows, in many cases, that these data be represented as vectors of a real vector space. For example, for images of N pixels, $\mathcal{X}$ is (a region of) $\mathbb{R}^N$ or $\mathbb{R}^{3N}$ depending on whether the images are monochrome or RGB colored. The dimension of these spaces, for images that are ordinarily of interest, is very large. In general, then, we must be prepared to work with spaces $\mathcal{X}$ of very high dimension.

Data Generation: Data generation is governed by the random selection of elements of $\mathcal{X}$ according to a probability distribution P over $\mathcal{X}$. This distribution is not known by the AL and is generally far from a uniform distribution. For example, the images we usually encounter present regularities that distinguish them from those formed by pixels selected at random according to a uniform distribution independently. A similar observation holds for sounds, such as music.

Hypothesis Space: This is a space $\mathcal{F}$ of functions $f : \mathcal{X} \to \mathbb{R}$ that we consider P-measurable. The selection of such a space is known as *inductive bias*, as it expresses a priori assumptions about the expected form of the solution distilled by the algorithm.

The space $\mathcal{F}$ is often specified as a space of parameterized functions. In the case of linear regression, the function space was that of multivariate polynomials of degree 1. In the following section, we will see that neurons and neural networks can be viewed as factories of non-linear parameterized functions, which is the most patent reason for their utility in AL.

Complexity: The measure of the *complexity* of the hypotheses is a function $\gamma : \mathcal{F} \to \mathbb{R}^+$. Example: $\gamma(f) = \|f\|$ (the norm of f) when $\mathcal{F}$ is a Banach space. For all $\delta \in \mathbb{R}^+$, we set $\mathcal{F}_\delta = \{f \in \mathcal{F} : \gamma(f) \leqslant \delta\}$. In the case of the norm, this is the (closed) ball of $\mathcal{F}$ of radius δ, which has the advantage of being a convex set. This notion allows to endow the learning algorithm with a criterion to grade the search effort according to increasing complexity. In the case of neural networks, this complexity is controlled by an implicit *penalization* through the optimization algorithm (Section 4 and section §5.4) or an explicit *regularization*, such as a penalty on the network weights.

Expert or Supervisor: It is a function $f^* : \mathcal{X} \to \mathbb{R}$. For each data point x, the expert produces an example: the pair (x, y), where $y = f^*(x)$. The

learning algorithm does not know the expert (which is why it is also called an *oracle*), but its purpose is to imitate it as faithfully as possible. If the expert is an element of $\mathcal{F}$ (which generally does not happen), we say it is *realizable*.

Training Data: It is a set of examples

$$\mathcal{D} = \{(x^i, y^i = f^*(x^i)) : x^i \in \mathcal{X}\}_{i\in[n]}$$

produced by the expert, where the x^i are generated according to the distribution P independently, $x^i \sim P$ in symbols.

Loss: The *loss* of $f \in \mathcal{F}$ is an indicator, denoted $L(f)$, of the separation between f and f^*. A common example is $\mathbb{E}_P|f(x) - f^*(x)|^2$, the expected value of $|f(x) - f^*(x)|^2$ relative to P. More generally, we can use $L(f) = \mathbb{E}_P\ell(f(x), f^*(x))$, where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a non-negative function with $\ell(y, y') = 0$ iff $y = y'$ (a *pointwise loss*). In some domains, the loss is called *risk* or *error*.

Objective of AL. While the primary objective is to approximate the expert f^*, in terms of the basic model it is a matter of finding an estimator $\hat{f} \in \mathcal{F}$, using only $\mathcal{D}$, such that its loss $L(\hat{f})$ is a good approximation of the *minimum loss* achievable with functions from $\mathcal{F}$, $L_{\mathcal{F}} = \min_{h\in\mathcal{F}} L(h)$. This estimator is built from an *empirical risk minimization* (ERM), where the *empirical risk* (or *empirical loss*) is the functional $\hat{L}_{\mathcal{D}} : \mathcal{F} \to \mathbb{R}^+$ defined by:

$$\hat{L}_{\mathcal{D}}(h) = \frac{1}{n}\sum_{i=1}^{n} |h(x^i) - y^i|^2.$$

It is an unbiased estimator of $L(h)$ through the data $\mathcal{D}$. Naturally, the expression $|h(x^i)-y^i|^2$ must be replaced by $\ell(h(x^i), y^i)$ if L is defined in terms of a pointwise loss ℓ.

Since there is a (random) discrepancy between the functional we are interested in minimizing (L, unknown) and the functional we at our disposal ($\hat{L}_{\mathcal{D}}$), it is necessary to introduce some type of regularization in order to control these fluctuations. For example, we can consider the δ-*restricted* empirical loss:

$$\hat{L}_{\mathcal{D},\delta} = \min_{h\in\mathcal{F}_\delta} \hat{L}_{\mathcal{D}}(h), \tag{3}$$

or the λ-*regularized* empirical loss (cf. the discussion of equation (2)), or λ-*penalized*,

$$\min_{h\in\mathcal{F}}(\hat{L}_{\mathcal{D}}(h) + \lambda\gamma(h)), \tag{4}$$

where $\gamma(h)$ is the complexity of h introduced on page 12

As we will see later, empirical risk minimization is achieved through methods known generically as *gradient descent algorithms*.

Error Decomposition. Given an estimator $\hat{f}$ obtained from ERM within a hypothesis space $\mathcal{F}$, the objective of statistical learning theory is to bound $L(\hat{f}) - L_{\mathcal{F}}$, a quantity that some authors call *regret*, as it expresses the discrepancy that an oracle knowing $L(\hat{f})$ and $L_{\mathcal{F}}$ would emit. This regret can be decomposed as follows [26, 10]:

$$L(\hat{f}) - L_{\mathcal{F}} = L(\hat{f}) - L_{\mathcal{F}_\delta} + L_{\mathcal{F}_\delta} - L_{\mathcal{F}}.$$

The significance of the term $L_{\mathcal{F}_\delta} - L_{\mathcal{F}}$ is described below as the *approximation error*, denoted ϵ_{app}. On the other hand, we can write:

$$L(\hat{f}) - L_{\mathcal{F}_\delta} = L(\hat{f}) - \hat{L}_{\mathcal{D}}(\hat{f}) + \hat{L}_{\mathcal{D}}(\hat{f}) - \hat{L}_{\mathcal{D},\delta} + \hat{L}_{\mathcal{D},\delta} - L_{\mathcal{F}_\delta}.$$

The difference $\epsilon_{\text{opt}} = \hat{L}_{\mathcal{D}}(\hat{f}) - \hat{L}_{\mathcal{D},\delta}$ is analyzed below as the *optimization error*. We are left to consider the sum $L(\hat{f}) - \hat{L}_{\mathcal{D}}(\hat{f}) + \hat{L}_{\mathcal{D},\delta} - L_{\mathcal{F}_\delta}$, for which we will provide an upper bound. On one hand, it is clear that $L(\hat{f}) - \hat{L}_{\mathcal{D}}(\hat{f}) \leqslant \epsilon_{\text{est}}$, defined as $\sup_{h\in\mathcal{F}_\delta} |L(h) - \hat{L}_{\mathcal{D}}(h)|$ and explained below as the *statistical* or *fluctuation error*; on the other hand, we also have $\hat{L}_{\mathcal{D},\delta} - L_{\mathcal{F}_\delta} \leqslant \epsilon_{\text{est}}$, since if $h \in \mathcal{F}_\delta$ satisfies $L_{\mathcal{F}_\delta} = L(h)$, then $\hat{L}_{\mathcal{D},\delta} - L_{\mathcal{F}_\delta} \leqslant \hat{L}_{\mathcal{D}}(h) - L(h) \leqslant |L(h) - \hat{L}_{\mathcal{D}}(h)| \leqslant \epsilon_{\text{est}}$. We can summarize these considerations in the following statement:

Theorem 2 (Regret bounding)**.** $L(\hat{f}) - L_{\mathcal{F}} \leqslant \epsilon_{\text{app}} + \epsilon_{\text{opt}} + 2\epsilon_{\text{est}}$

Approximation error: We have defined it as the difference $\epsilon_{\text{apr}} = L_{\mathcal{F}_\delta} - L_{\mathcal{F}}$. It does not depend on the data $\mathcal{D}$, is nonnegative, and decreases as δ increases. It measures the approximation of the minimum defect $L_{\mathcal{F}}$ that can be achieved with functions of $\mathcal{F}_\delta$. For a discussion on the design of spaces $\mathcal{F}$ that allow approximating f^*, see §3.

Optimization Error: For $h \in \mathcal{F}_\delta$, we have defined it as $\epsilon_{\text{opt}} = \hat{L}_{\mathcal{D}}(h) - \hat{L}_{\mathcal{D},\delta}$, where $\hat{L}_{\mathcal{D},\delta}$ is the minimum of the empirical losses of functions in $\mathcal{F}_\delta$. In practice, a tolerance $\epsilon > 0$ is set and an empirical risk minimization algorithm is applied to obtain $\hat{f} \in \mathcal{F}_\delta$ such that its optimization error is small, that is:

$$\hat{L}_{\mathcal{D}}(\hat{f}) - \hat{L}_{\mathcal{D},\delta} \leqslant \epsilon. \tag{5}$$

The main problem in achieving (5) is computational, due to the non-convex landscape of the empirical risk, and we study it in detail in §4.

Statistical Error: For an $h \in \mathcal{F}_\delta, |L(h) - \hat{L}_{\mathcal{D}}(h)|$ is the error committed by substituting the loss $L(h)$ with the empirical loss $\hat{L}_{\mathcal{D}}(h)$. In the worst case, this error is $\epsilon_{\text{est}} = \sup_{h\in\mathcal{F}_\delta} |L(h) - \hat{L}_{\mathcal{D}}(h)|$, and we call this quantity the *statistical error* or *fluctuation error*. We analyze its role in the analysis of generalization capability in §5.

Theorem 2 tells us that we can guarantee good learning (that is, $L(\hat{f}) - L_{\mathcal{F}}$ is small and, therefore, $\hat{f}$ is a good approximation of f^* with a function from $\mathcal{F}$) if

we can ensure that the three error terms are small. The requirements to achieve this: to choose a good $\mathcal{F}$; to find a good optimization algorithm that warrants inequality (5); and to devise tools to control the fluctuations that produce the statistical error. That is, we ask ourselves:

(1) How can we find hypothesis spaces $\mathcal{F}$ with good approximation properties in high dimensions?

(2) What algorithms are available to achieve the minimization of $\hat{L}_{\mathcal{D}}(h)$?

(3) How can statistical fluctuations be controlled?

We dedicate §3, §4, and §4 to the study of these questions, respectively.

2.3 Neurons and Neural Networks

In this section, we introduce the neuron as the basic tool for constructing functional spaces with good approximation and generalization properties in high dimensions.

In AL, a *neuron* (see Figure 4*a*)) is a function of the form:

$$x \mapsto \chi(\theta, x) = a\sigma(x \cdot w + w_0), \tag{6}$$

where $a, w_0 \in \mathbb{R}$, $w \in \mathbb{R}^d$, $\theta = (a, w, w_0)$, and where σ is a *sigmoid-shaped* function, such as the *logistic function* $\sigma(t) = (1 + e^{-t})^{-1}$ (Figure 2). We say that θ are the *parameters* of the neuron. More specifically, w is the *weight vector* and w_0 is the *bias*. This model computes, for each value of the parameters θ, a function whose graph has the shape of a *ridge* (see Figure 4*b*)), since it is constant over each hyperplane perpendicular to the vector w and has a sigmoid shape in the direction of w.

By adding the constant $x_0 = 1$ as an extra input, and taking $\tilde{x} = (1, x)$ and $\bar{w} = (w_0, w)$, we have that $x \cdot w + w_0 = \tilde{x} \cdot \bar{w}$, so that if $a = 1$ and σ is the logistic function, the neuron computes the logistic regression of $\tilde{x}$ with weights $\bar{w}$.

If the neuron represented in Figure 4*a*) is composed with the function $s \mapsto \pm 1$ depending on whether $s = f_\theta(x) \geqslant 1/2$ or $s < 1/2$, we essentially have the so-called *Rosenblatt perceptron* (1958), which historically is significant for being the first system that could learn a binary classification from examples through a process of adjusting the parameters θ.

A *neural network* (NN) can be thought of as a composition of neurons according to a connection graph known as the *architecture* of the network. Here we will focus primarily on the case of directed graphs, leaving for later some comments on undirected networks, such as *Hopfield networks* or so-called *Boltzmann machines*. We will also not consider networks with feedback —that is, those containing closed paths.

The standard architecture of a *feed-forward* NN is a directed graph structured in layers L_j, as in Figure 5, and its functional signature can be condensed into the following scheme:

$$\mathcal{N} : \text{Input } L_0 \to L_1 \to L_2 \to \cdots \to L_m \to L_{m+1} \text{ Output.} \tag{7}$$

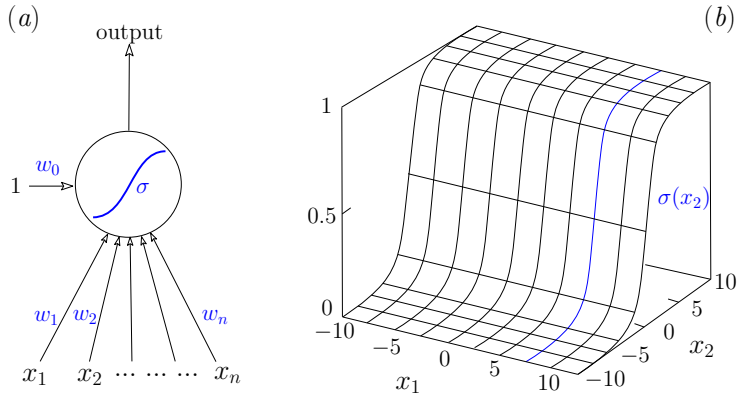

Figure 4: (a) Schematic of a neuron. (b) The graph of the function calculated by a neuron has the shape of a ridge. Functionally, we can consider that the weights w_j and the parameters of σ, collectively θ, form part of the neuron, and that it outputs $f_\theta(x_1, \dots, x_n)$. Graphically, we represent this functionality with a circle.

Conventionally, a network is considered *shallow* if $m = 1$ and *deep* if $m > 1$. Layers $L_1, \dots, L_m$ are considered *hidden*, while the input and output layers are *visible*. The *width* of a layer is the number of neurons it contains.

Each layer consists of a certain number of neurons, and there are only connections from the neurons of one layer toward neurons of the next layer. Each layer receives a signal x from the previous layer and produces an output signal x' that it sends to the next layer, which we can represent as a functional relation $x' = f_j(x)$. The layer L_0 collects the input signal (of a sound or an image, for example), a role that presents a certain analogy with that of the sensory organs of living beings. The output, represented by layer L_{m+1}, is the result of progressively applying (feed-forward) the functions $f_1, f_2, \dots, f_m$ (that is, the composition $f_m \circ f_{m-1} \circ \cdots \circ f_1$ to the input. The output, represented by layer L_{m+1}, is the result of progressively applying (feed-forward) the functions $f_1, f_2, \dots, f_m$ (that is, the composition $f_m \circ f_{m-1} \circ \cdots \circ f_1$) to the input. Following the biological analogy, the output can be the signal sent to various systems of the living being, such as the locomotor system, or the phonation organs of humans, among many others.

The function $f_j : x \mapsto x'$ depends on the set of parameters (or *weights*) associated with the neural connections reaching L_j, so that f_j is a parameterized function that we can denote as f_{θ_j}, and thus the progressive action of the network is the parameterized function $f_\theta = f_{\theta_m} \circ \cdots \circ f_{\theta_1}$ $(\theta = (\theta_1, \dots, \theta_m))$. Given that the activation functions of the neurons are non-linear, this function is also non-linear. The total number of parameters in a neural network is generally very large, especially in deep ones. To the extent that the biological analogy may hold, the parameters of the network play the role of synaptic connections, of which it is estimated there are, in the case of human brains, tens of trillions.

The character of a layer L_j of the NN (7) depends on how the parameters

θ_j are used in the definition of f_{θ_j}. An important case is that of *convolutional* layers, a generic term to designate that in the definition of $f_{\theta_j}(x)$, some kind of *convolution* or *cross-correlation* is used between θ_j and x; in this case, the parameters θ_j play the role of *filters*. A NN is considered *convolutional* (CNN) if it contains at least one convolutional layer.

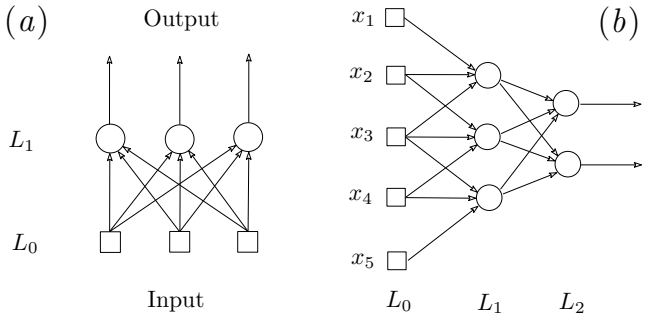

Figure 5: (a) NN with no hidden neurons and fully connected. (b) A NN with a hidden layer of three neurons, L_1, fully connected to the output neurons, L_2. The input layer, L_0, is only partially connected to layer L_1. Since each hidden neuron receives outputs from three input neurons, the weights of the three hidden neurons could be the same, in which case they are referred to as shared weights. This idea is the basis for convolutional networks, which we consider later. If the three weights are w_1, w_2, w_3, the state of the hidden neurons is $y_j = \sum_{i=1}^{3} w_i x_{i+j-1}$ for $j = 1, 2, 3$; that is, the vector of these states is the convolution of the input vector x with the weight vector w.

2.4 Universal Approximators

Standard neural networks have the capacity to approximate any measurable function to any desired precision. In fact, this can be achieved with shallow networks, but with a number of hidden neurons that increases as more precision is required.

Theorem 3 ([49, 91, 15, 156])**.** *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a continuous, bounded, monotonically increasing, and non-constant function* (e.g., a sigmoid or tanh). *Then the space $\mathcal{F}$ of functions of the form*

$$\sum_{j=1}^{N} a_j \sigma(w_j \cdot x + b_j), \tag{8}$$

where $N \in \mathbb{Z}^+$, $w_j \in \mathbb{R}^n$, $a_j, b_j \in \mathbb{R}$, is dense (uniform convergence on compact sets) *in the space of continuous functions of $\mathbb{R}^n$. In particular, it follows that any continuous function of $\mathcal{X}$ can be approximated* (uniformly on compact sets) *as closely as desired by a shallow neural network.*

See [156, proposition 6.4] for a derivation within the framework of the Stone-Weierstrass theorem. This result gives confidence that neural networks can approximate any function of practical interest, but in itself, it is not a quantitative result, as it does not incorporate, for example, any control over the number of neurons in the hidden layer required to achieve a certain approximation error. Recall from section §2.2 that approximation is only one piece of the puzzle, and that a good learning algorithm also needs a good generalization and optimization rate—properties that, in practice, require deeper architectures.

3 Approximation

A fundamental problem is guaranteeing that learning is feasible even when $n \ll \epsilon^{-d}$, a situation where classical methods are not applicable (§3.1), but where neural networks offer some hope. To address this issue, we will first look at the role of depth (§3.2) and then at the spatial geometry induced by the physics of the problem (§3.3).

3.1 The Dimensional Curse of Approximation

The analysis of functions $f : \Omega \to \mathbb{R}$ defined in a domain Ω of low dimension is a mature discipline with a long history in mathematics. The regularity of a function in one, two, or three dimensions can be precisely characterized to express the structure of a specific problem; for example, the solutions of a wave equation, or an image $f(u)$, $u \in [0,1]^2$, using harmonic analysis tools (such as the Fourier transform or wavelets), or geometric variants (such as ridgelets). This control allows for the resolution of all kinds of inverse problems (such as image compression, magnetic resonance, etc.) with statistical and computational guarantees. In these applications, we can say that the mathematical models are "up to date" regarding the algorithms used in practice.

The situation in high dimensions is totally different: functional spaces based on global notions of regularity, such as Sobolev spaces, lack adaptability as the dimension grows: a classic result in non-parametric statistics [197] establishes that estimating a function f of the Sobolev class in dimension d with least squares error ϵ requires $O(\epsilon^{-2-d/s})$ samples, where s is the number of derivatives of f. In other words: only extremely regular functions, with millions of finite derivatives, are estimable in the high-dimensional regime! Local notions of regularity, such as simply Lipschitz functions, also suffer from the dimensional curse, as ϵ^{-d} samples are necessary to obtain minimax errors of order ϵ [206].

It is necessary, then, to work with an alternative approximation theory to respond to the challenge of high dimensions. In this context, functional spaces associated with neural networks, starting with the shallow functions described by equation (8), have been studied by great analysts (Pinkus, Matusek, or even Bourgain) as generalizations of integral representations in Fourier analysis:

$$f(x) = \int g(\theta)\sigma(x \cdot \theta)d\theta, \tag{9}$$

where g is the "transform" and σ is a generic non-polynomial activation that generalizes the complex exponential $\sigma(x \cdot \theta) = e^{-i(x \cdot \theta)}$ of the Fourier transform. An important result of [10] is that these functional spaces allow for *adaptation* to low-dimensional structures without the curse: if $f(x) = \tilde{f}(Ux)$, where $U \in \mathbb{R}^{k \times d}$ with $k \ll d$ is an unknown projection operator, the regularity of f is expressed in terms of $y = Ux \in \mathbb{R}^k$, which has a dimension k potentially much lower than the ambient space dimension d. The essence of these spaces is, once again, the *sparsity* that can be imposed on the representation (9), in line with fundamental results from the 1990s and 2000s [55] that transformed signal processing using the sparsity condition.

3.2 From Shallow Networks to Deep Convolutional Networks

The preceding mathematical approximation arguments must be contrasted with experimental arguments, which show an enormous difference between generic shallow networks and modern convolutional networks with hundreds of layers. It can be said that the harmonic analysis of deep networks is still in its infancy. In fact, the most profound results on this subject come from the theoretical computer science community, which has studied logical circuit approximation questions since the seventies based on harmonic analysis of the hypercube [118].

In the context of neural networks, some authors have studied the role of depth from the point of view of approximation. We cite in particular [58], where the authors present a class of radial and oscillating functions in dimension d that require an exponential number of neurons to be approximated by shallow networks, which provides a negative counterpoint to the universal approximation result of Theorem 3, but at the same time they show that they can be satisfactorily approximated with deep networks.

In addition to depth, another crucial aspect of modern neural networks is that they incorporate knowledge of physics (in the spatial structure of images, videos, and sounds, for example), as we describe below.

3.3 The Role of Spatial Geometry

Up to now, we have proceeded thinking that $\mathcal{X}$ is $\mathbb{R}^d$, $d \gg 1$, but in the most important applications (such as computer vision, artificial language—text and speech—or physics) we have more structure. In many cases, the elements of $\mathcal{X}$ are functions $x : \Omega \to \mathbb{R}^k$, where Ω is a region of a low-dimensional space and k is a small positive integer. In the case of an image, Ω can be a rectangle in the plane and k the number of channels, i.e., $k = 1$ for monochrome images or $k = 3$ for color images. In the case of speech, Ω can be $\mathbb{R}$, so that $x(t)$ is the sound intensity at instant t, with $k = 2$ if the sound is stereo. In the case of physics of N particles, $\Omega = (\mathbb{R}^3 \times \mathbb{R}^3)^N$, where the i-th particle is determined by the pair $(p_i, q_i) \in \mathbb{R}^3 \times \mathbb{R}^3$ formed by its linear momentum p_i and its position q_i. In all these cases, the high dimension of the observations x conceals the underlying low-dimensional structure, and the key question is how

we can use this structure (which we will call the *functional structure* of $\mathcal{X}$) for the treatment of the aforementioned problem.

Global Symmetries. In AL, symmetries appear as transformations of the inputs that do not affect the function we want to learn. This is a situation that presents an analogy with many in mathematics and physics. For example, the relationship between the Galois group of a polynomial and the structure of its roots, or between the symmetries of a physical system and the conservation laws that govern it (Noether's theorems).

A symmetry of $\mathcal{F}$ is an invertible transformation $T : \mathcal{X} \to \mathcal{X}$ such that $f(x) = f(Tx)$ for any $f \in \mathcal{F}$. Of these symmetries, which form a group G with the operation of composition, the most significant for our purposes are those induced from symmetries of the functional structure. More specifically, if $\tau : \Omega \to \Omega$ is a transformation, then we can consider $T_\tau : \mathcal{X} \to \mathcal{X}$, $x \mapsto x_\tau$, defined by the relation $x_\tau(u) = x(\tau(u))$. For example, in the case of a translation $\tau = t_v$, $t_v(u) = u + v$, we have $x_{t_v}(u) = x(u + v)$.

However, the global invariance assumption is not strong enough to steer estimation in high dimensions. Transformation groups are typically low-dimensional; in signal processing, they are often subgroups of the affine group $\mathrm{Aff}(\Omega)$, which has dimension 6 in the case of images. A much stronger premise can be defined by specifying how f behaves with respect to geometric perturbations of Ω close to elements of these global symmetry groups.

Local Deformations and Scale Separation. Consider now a vector field $\tau : \Omega \to \mathbb{R}^d$, sufficiently regular, which produces a local deformation $\varphi(u) := u + \tau(u)$, acting on $L^2(\Omega)$ by composition, $x_\tau(u) := x(\varphi(u))$. Examples include local translations, changes in viewpoint, rotations, and frequency transpositions [33], which have been used extensively as models of image variability in computer vision [96, 62, 70]. Note that if $\|\tau\| < 1$, then φ is a diffeomorphism.

Most tasks in computer vision are stable under deformations, in the sense that the prediction made by f does not change much if the input image is slightly deformed [33, 122]. In tasks that are invariant under translation, this premise can be expressed as

$$|f(x_\tau) - f(x)| \approx \|\tau\|, \tag{10}$$

for all x and all τ, where $\|\tau\|$ measures the distance of φ to the translation group; for example, $\|\tau\| := \sup_u \|\nabla\tau(u)\|$, [33]. Stability under deformations is a strong premise because the space of local deformations has high dimension, as opposed to the global symmetry group, as illustrated in Fig.6. Additionally, this stability under deformations can be expressed in terms of a *multiresolution analysis* (MRA) [120] as follows. A continuous image x defined on Ω can be interpreted as a linear combination $x = x_J + \sum_{j<J} \tilde{x}_j$ of signals $\tilde{x}_j$, $j \in (-\infty, J)$, where x_J captures large-scale structures 2^J and $\tilde{x}_j$, *details* at smaller scales 2^j, as illustrated in Figure 7. Multiresolution analysis constructs linear operators based on *wavelets* to obtain this decomposition $x \mapsto \{x_J, \tilde{x}_j\}$. Similarly, functions f of interest in vision admit an approximate factorization in terms of a

multiresolution analysis: denoting by P_j a projector (not necessarily linear) on the space of signals at scale at least j, the geometric stability of f is expressed as

$$f(x) \approx f_j(P_j(x)),$$

where f_j is a new function defined on images at a lower resolution, limited to scales larger than j. In other words, interactions between pixels can be "separated" according to scales by first treating local interactions through the operator P_J. Applying this factorization iteratively to $f_j, f_{j+1}, \ldots$, we obtain a separation of learning at each scale. This principle of scale separation appears in various fields of mathematical physics, for example, in the *fast multipole method* [73]. This principle of stability under local deformations has been exploited in computer vision for various models, including convolutional networks, and analyzed mathematically with the *scattering transform* [33, 122].

Figure 6: Deformations of an image x by a regular field τ do not alter the semantic information x_τ. However, strong deformations (right-most image) can do so.

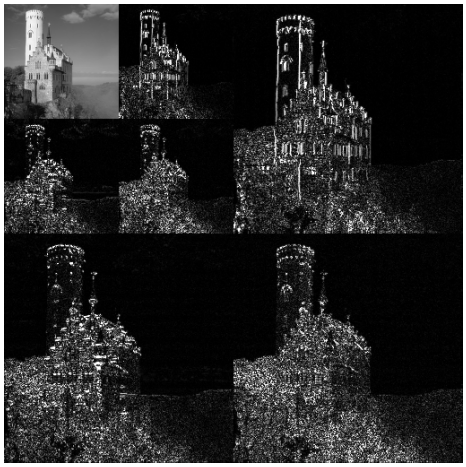

Figure 7: Multi-resolution decomposition. Source: Wikipedia.

3.4 Representations with Geometric Stability

Motivated by the premise of geometric stability, we are interested in constructing signal representations that are compatible with it. Suppose, for concrete purposes, that the estimate $\hat{f}$ of f, the target function, has the form:

$$\hat{f}(x) := \Phi(x) \cdot w, \tag{11}$$

where $\Phi : L^2(\Omega) \to \mathbb{R}^K$ designates the representation of the signal and $w \in \mathbb{R}^K$ the classification or regression coefficients. In a CNN, Φ would be the operator that projects the input signal to the last level of its scattering representation, and w is associated with the last output signal of the network. The scattering representation is used in harmonic analysis to extract information from signals with certain guarantees of invariance and stability under deformations. The central idea is to iteratively apply a wavelet transform with a non-linear transformation (*rectification*), which allows for capturing interactions between different scales of the transform (hence the name "scattering"). The linear relationship (11) between $\Phi(x)$ and $\hat{f}(x)$ implies that geometric stability in the representation is sufficient to guarantee a predictor that is also geometrically stable. Indeed, if we assume:

$$\forall x, \tau, \|\Phi(x) - \Phi(x_\tau)\| \lesssim \|x\|\|\tau\|, \tag{12}$$

then, by Cauchy-Schwarz:

$$|\hat{f}(x) - \hat{f}(x_\tau)| \leqslant \|w\|\|\Phi(x) - \Phi(x_\tau)\| \lesssim \|w\|\|x\|\|\tau\|.$$

This motivates the study of signal representations that satisfy (12), while simultaneously capturing enough information to ensure that $\|\Phi(x) - \Phi(x')\|$ is large if $|f(x) - f(x')|$ is also large. In this context, a notable challenge in order to achieve both properties simultaneously involves transforming the high-frequency components of x in a stable manner.

In recognition tasks, besides geometric stability, it is natural to demand stability with respect to the $L^2(\Omega)$ metric:

$$\forall x, x' \in L^2(\Omega), \|\Phi(x) - \Phi(x')\| \lesssim \|x - x'\|. \tag{13}$$

This stability property ensures that the presence of additive noise in the input signal will not drastically change the characteristics of the representation.

The two desired stabilities, (12) and (13), can also be interpreted in terms of robustness against so-called *adversarial examples* [191]. Indeed, the generic context of adversarial examples consists of producing small perturbations x' of a given signal x (measured by convenient norms) so that $|(\Phi(x) - \Phi(x')) \cdot w|$ is large. The stability of the representations means that these adversarial examples cannot be obtained with small additive or geometric perturbations.

In summary, the stability properties (12) and (13) can be taken as axioms to define adapted function spaces. The scattering transform ([33, 121]) defines a space of linear functions using equation (11), which can be generalized thanks to the theory of reproducing kernels [24] with the same stability principles. We

briefly mention that these geometric stability principles can be generalized to non-Euclidean domains, thanks to intrinsic tools in harmonic analysis, Riemannian manifolds, and graphs, in addition to group representation theory, giving rise to so-called *geometric deep learning*; see [31] for a detailed summary of these types of generalizations. Although this axiomatic view allows for good numerical results with mathematically rigorous methods, we emphasize in conclusion that it defines (linear) approximation spaces much smaller than those defined by deep neural networks, and understanding this contrast is one of the most important open problems in the mathematics of AL.

4 Optimization

Beyond the open questions described in the previous section, possibly the thorniest mathematical aspect of the AL puzzle is computational: how can we define algorithms with guarantees for empirical risk optimization when the class of functions is non-linear? To this end, we review the gradient descent method in §4.1, followed by explaining first the connection with reproducing kernels in §4.2 and then with the dynamical systems of measure transport in §4.3.

4.1 Gradient Descent Method

Recall from the basic model that the objective of supervised learning consists of minimizing the empirical risk in a hypothesis class with limited capacity. In penalized form, this translates to (cf. equation (4)):

$$\min_{h\in\mathcal{F}} E(h), \quad E(h) = L_{\mathcal{D}}(h) + \lambda\gamma(h). \tag{14}$$

Here we are formulating the learning problem *implicitly*, directly seeking functions from class $\mathcal{F}$, for example, the functions produced by a neural network with a predetermined architecture.

Problem (14) is conceptually simple but computationally complicated, as in practice the space $\mathcal{F}$ often has a non-Euclidean geometry. If we think of $\mathcal{F}$ as a subspace of the space of *all* functions $f : \mathcal{X} \to \mathbb{R}$, we can interpret (14) geometrically as the minimization of a convex cost in an arbitrary domain.

Although it is possible to define iterative methods to optimize functions at this level of generality, in our case we will simplify the problem by introducing a *parameterization* of the hypothesis space $\phi : \Theta \to \mathcal{F}$, where Θ is a Euclidean domain, and where we assume that it is differentiable and surjective. In the case of a neural network, $\boldsymbol{\theta} \in \Theta$ represents the parameters to be adjusted, and $\phi(\boldsymbol{\theta})$ is the function implemented by the network corresponding to the parameters $\boldsymbol{\theta}$. This parameterization allows us to define a Euclidean optimization domain, at the price that now the function to be optimized is generally non-convex:

$$\min_{\boldsymbol{\theta}\in\Theta} \tilde{E}(\boldsymbol{\theta}), \quad \tilde{E}(\boldsymbol{\theta}) := E(\phi(\boldsymbol{\theta})). \tag{15}$$

The classical strategy for attacking (15) is the gradient descent method (Cauchy, 1847). Starting from an arbitrary point $h_0 = \phi(\boldsymbol{\theta}_0) \in \mathcal{F}$, we consider the iteration:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \nabla \tilde{E}(\boldsymbol{\theta}_k), \tag{16}$$

where $\nabla \tilde{E}$ is the first variation or gradient of $\tilde{E}$ and $\eta > 0$ is an adjustable parameter corresponding to the optimization step. We can interpret (16) geometrically using the linear approximation of $\tilde{E}$ in a neighborhood of $\boldsymbol{\theta}_k$. Assuming $\tilde{E}$ is β-smooth and $\eta < \beta^{-1}$, we have the upper bound:

$$\tilde{E}(\boldsymbol{\theta}) \leqslant \tilde{E}(\boldsymbol{\theta}_k) + \nabla \tilde{E}(\boldsymbol{\theta}_k) \cdot (\boldsymbol{\theta} - \boldsymbol{\theta}_k) + \eta^{-1} \|\boldsymbol{\theta} - \boldsymbol{\theta}_k\|^2,$$

which leads us to rewrite (16) in variational form:

$$\boldsymbol{\theta}_{k+1} = \operatorname{argmin}_{\boldsymbol{\theta}} \left[\tilde{E}(\boldsymbol{\theta}_k) + \nabla \tilde{E}(\boldsymbol{\theta}_k) \cdot (\boldsymbol{\theta} - \boldsymbol{\theta}_k) + \eta^{-1} \|\boldsymbol{\theta} - \boldsymbol{\theta}_k\|^2 \right]. \tag{17}$$

The algorithm exploits the local regularity of $\tilde{E}$ to define a quadratic model that upperbounds the function, and to find a point in the Euclidean neighborhood of the current parameter where the error can be decreased. The gradient descent method admits several generalizations that we will not enter into in this article; for example, exploiting second-order approximations or generalizations in non-Euclidean metrics. It should be mentioned that in the convex case, the gradient descent method enjoys a precise mathematical analysis, which allows guaranteeing that the method, in its accelerated version [138], is optimal with respect to an oracle complexity model that uses linear combinations of past gradients [137], in the sense that the convergence rate $\Theta(1/t^2)$ cannot be improved. For more details on the method, we refer the reader to the excellent monograph [34].

In our context, the crucial result is that the gradient descent method allows finding local minima of $\tilde{E}$ without the dimensional curse: for a given error $\epsilon > 0$, $\tilde{O}(\epsilon^{-2})$ gradient descent iterations are required (appropriately modified in the presence of noise) to find an approximate local minimum within ϵ, where $\tilde{O}(f)$ means $\mathcal{O}(f)$ ignoring logarithmic factors, [97].

Despite this robust behavior against the dimensional curse, the gradient descent method, as described in (16), is problematic in large-scale applications, since each parameter update requires the gradient of the empirical risk $\tilde{E}$, which depends on all the data. This has motivated fruitful research into stochastic gradient methods, beginning with the seminal work of Robbins and Monro in the fifties [163].

Without entering into technical details, the stochastic gradient method replaces the gradient oracle with a stochastic oracle: given the current point $\boldsymbol{\theta}$, the algorithm has access to a random vector $\boldsymbol{\theta}'$ such that the conditional expectation $\mathbb{E}(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \nabla \tilde{E}(\boldsymbol{\theta})$; in other words, we have access to an unbiased estimator of the gradient of the function. In the case of empirical risk, this estimator is obtained by evaluating the risk at a single point, $\nabla R_i(\boldsymbol{\theta})$, where $R_i(\boldsymbol{\theta}) = |\phi(\boldsymbol{\theta}, x_i) - y_i|^2 + \lambda\gamma(\phi(\boldsymbol{\theta}))$, where $\phi(\boldsymbol{\theta}, x_i) = \phi(\boldsymbol{\theta})(x_i)$, or on a small subset of points (mini-batch). Under these conditions, it is possible to establish

guarantees on iteration complexity similar to those of the deterministic case, on the order of $\tilde{O}(d\epsilon^{-4})$ stochastic iterations, to find an approximate local minimum within ϵ, see [97].

We mention in particular that, in the case of NNs, a popular variant of the gradient descent method considers adaptive gradient steps, where the hyperparameter η is replaced by a step adapted to each coordinate of $\boldsymbol{\theta}$. Among the various alternatives, the most popular is *Adam* [100], where the gradient step is normalized in each coordinate based on an estimate of the norm and combined with a momentum component.

In summary, in the regime where we manipulate hypothesis spaces with enormous degrees of freedom, with no structure other than the local regularity of the parameterization, the stochastic gradient descent method emerges as the canonical tool for learning, with an iteration complexity redeemed from the dimensional curse. However, this analysis only guarantees that it is relatively easy to find a local minimum, not a global minimum. Is it possible to certify the global convergence of empirical risk in the context of neural networks? In the following section, we will see that, in some cases, thermodynamics can provide positive answers.

4.2 Ultra-parameterized NNs and Tangent Kernels

The gradient descent method (16) can be interpreted as an Euler discretization of an ordinary differential equation:

$$\dot{\boldsymbol{\theta}} = -\nabla \tilde{E}(\boldsymbol{\theta}),$$

with a random initial condition $\boldsymbol{\theta}(0)$ given by a certain probability distribution defined over the parameter space Θ.

This equation, called *gradient flow*, defines a continuous dynamics $\boldsymbol{\theta}(t) \in \Theta$, $t \geqslant 0$, in the parameter space, which in turn generates a dynamics in the functional space $\mathcal{F}$ defined by $h(t) = \phi(\boldsymbol{\theta}(t))$; see Figure 8. The chain rule immediately implies an equivalent gradient flow in $\mathcal{F}$, given by:

$$\dot{h} = -\mathcal{K}(t) \cdot \nabla E(h(t)), \tag{18}$$

where $\mathcal{K}(t) := D\phi(\boldsymbol{\theta}(t))^\top D\phi(\boldsymbol{\theta}(t))$ is a change of metric called the *tangent kernel*, which projects the functional gradient $\nabla E(h(t))$ onto the tangent space of the manifold $\mathcal{F}$, and where $D\phi(\boldsymbol{\theta})$ is the derivative of ϕ at point $\boldsymbol{\theta}$. The difficulty in analyzing (18) mathematically lies in the temporal variation of the tangent kernel $\mathcal{K}(t)$.

In fact, the temporal variation of $\mathcal{K}(t)$ can be understood geometrically as the curvature of the function space $\mathcal{F}$ in a neighborhood of the point $h(t)$. In the case where $\phi(\boldsymbol{\theta})$ is a neural network as in equation (7), an important result [95, 57, 56, 45] is that this curvature progressively disappears as the network becomes wider, under a certain normalization of the weights; therefore, $\mathcal{K}(t) \to \mathcal{K}(0)$ uniformly in finite time [95]. For example, in the case of a shallow

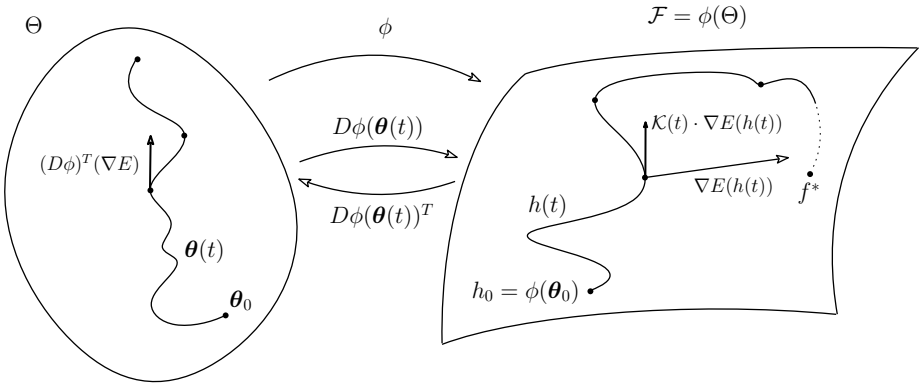

Figure 8: Gradient descent algorithm: geometric and analytical relationships between the parameter space Θ and the hypotheses space $\mathcal{F}$. In general, f^* lies outside $\mathcal{F}$.

network, if we consider:

$$\phi(\boldsymbol{\theta}, \cdot) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} \chi(\theta_j, \cdot),$$

where $\chi(\theta, \cdot)$ is the neuron defined in (6), and the parameters θ_j are considered independent and identically distributed according to a distribution μ_0, the tangent kernel becomes:

$$\begin{aligned} \mathcal{K}(t)[x, x'] &= \frac{1}{m} \sum_{j=1}^{m} \nabla_\theta \chi(\theta_j(t), x) \nabla_\theta \chi(\theta_j(t), x') \\ &\to \mathbb{E}_{\theta \sim \mu_0}[\nabla_\theta \chi(\theta, x) \nabla_\theta \chi(\theta, x')] \\ &:= \bar{\mathcal{K}}(x, x') \quad (m \to \infty). \end{aligned}$$

Under fairly general conditions [160], the tangent kernel at a finite width m concentrates uniformly toward $\bar{\mathcal{K}}$, with fluctuations of order $\sim 1/\sqrt{m}$. In this asymptotic regime $m \to \infty$, and considering for simplicity the non-regularized case $E(h) = \frac{1}{2}\|h - f^*\|^2$, the training dynamics simplifies to:

$$\dot{f} = -\bar{\mathcal{K}} \nabla E(f(t)) = -\bar{\mathcal{K}}(f(t) - f^*),$$

which corresponds to the linear dynamics associated with a linear regression model in a Hilbert space generated by the *reproducing kernel* $\bar{\mathcal{K}}$. These functional spaces (RKHS) are generalizations in infinite dimensions of Euclidean spaces and they allow for an in-depth and precise study of approximation, generalization, and optimization, though they suffer from the dimensional curse explained in §3.1.

With this parameterization, wide neural networks behave like linear models characterized by a tangent kernel that remains constant. While this explains

the good behavior of the gradient descent method, it does not explain mathematically the advantages of the non-linear models defined by networks, nor does it avoid the curse of dimensionality. As we will see, it is possible to capture the non-linear aspects with a convex model by changing the normalization.

4.3 Thermodynamic Limits and Shallow Networks as Particle Systems

Let us now look at the case of shallow networks (a single hidden layer) and consider a new normalization (in $1/m$ instead of $1/\sqrt{m}$) of the weights:

$$\phi(\boldsymbol{\theta}, x) = \frac{1}{m} \sum_{j=1}^{m} \chi(\theta_j, x), \tag{19}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_m)$ and the neuron $\chi(\theta, \cdot) : \mathcal{X} \to \mathbb{R}$ is a function parameterized by $\theta \in \Omega \subseteq \mathbb{R}^m$. For example, for $\chi(\theta, x) = a\sigma(x \cdot b + c)$, $\theta = (a, b, c) \in \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}$. Thanks to the simple structure of shallow networks, we can think of the function as an average of m simple functions $\chi(\theta_j)$, parameterized by m particles $\theta_1, \ldots, \theta_m \in \Omega$. The parametric view of a neuron network in terms of its weights (or particles) corresponds in the language of fluid mechanics to a Lagrangian model of the system, which contrasts with the Eulerian model, which expresses the system in terms of the *particle density* μ, defined as a probability measure over the space Ω. Indeed, we can rewrite (19) as:

$$\phi(\mu, x) = \int_{\Omega} \chi(\theta, x) \mu(d\theta), \tag{20}$$

taking $\mu = \mu^{(m)}$ as the empirical measure associated with the m particles:

$$\mu^{(m)}(d\theta) = \frac{1}{m} \sum_{j=1}^{m} \delta_{\theta_j}(d\theta).$$

The Eulerian perspective amounts to an abstraction of the discrete aspect of shallow networks, in the sense that: training is not about understanding the trajectory of each neuron, but the trajectory of the joint function they define for a sufficiently large number m. Mathematically, this corresponds to the canonical example of concentration of measure, the law of large numbers, which says that if each neuron θ_j is initialized independently from a law μ_0, then the empirical measure $\mu^{(m)}$ converges weakly to μ_0 with rate $1/\sqrt{m}$. Thus, a shallow network initialized with m particles $\theta_j \sim \mu_0$ approximates the *mean field* function $\phi(\mu_0, x) = \mathbb{E}_{\theta \sim \mu_0} \chi(\theta, x)$, with fluctuations of order $1/\sqrt{m}$:

$$\mathbb{E}\left[\int g(\theta) \left(\mu^{(m)}(d\theta) - \mu_0(d\theta) \right) \right] \lesssim \frac{1}{\sqrt{m}}, \tag{21}$$

where the expectation is with respect to the empirical measure, and g is a Lipschitz test function. In other words, a shallow network $\phi(\boldsymbol{\theta}, \cdot)$, initialized

with m independent and identically distributed particles $\theta_j \sim \mu_0$, approximates the *mean field* function $\phi(\mu_0, x) = \mathbb{E}_{\theta\sim\mu_0}\chi(\theta, x)$, with typical fluctuations of the order $1/\sqrt{m}$, uniformly in x over compact sets. This is nothing more than the classic Monte Carlo estimator. Next we will see how this sampling perspective also allows for the incorporation of the dynamic aspect of training.

The density allows for a linear parameterization of functions $\mathcal{F}$. Consequently, the Eulerian alternative to (15) is:

$$\min_{\mu\in\mathcal{P}(\Omega)} \mathcal{E}[\mu] := E(\phi(\mu, \cdot)), \tag{22}$$

where $\mathcal{P}(\Omega)$ denotes the space of probability measures on Ω. Thanks to the original convexity of E and the linearity of (20), this problem is convex with respect to μ, contrarywise to (15).

Nevertheless, the optimization of problem (22) is not trivial because the geometry of $\mathcal{P}(\Omega)$ giving rise to this convexity (the geometry of *mixtures*, in which the midpoint between two measures μ and ν is the mixture $\frac{1}{2}(\mu + \nu)$) is not compatible with the gradient descent implemented in practice. Specifically, the gradient descent algorithm, acting on the parameters of each particle, defines a proximal operator analogous to that of (17), but with a *transport* metric:

$$\mu_{k+1} = \operatorname{argmin}_\mu\{\mathcal{E}[\mu_k] + \delta\mathcal{E}[\mu_k]\cdot(\mu - \mu_k) + \eta^{-1}W_2(\mu, \mu_k)\}, \tag{23}$$

where $W_2(\mu, \mu')$ is the 2-Wasserstein distance between measures μ and μ' and $\delta\mathcal{E}$ is the first variation of $\mathcal{E}$ with respect to μ. Similarly to the Euclidean case, this proximal step admits a limit in continuous time, but in this case, instead of an ordinary differential equation, we obtain a partial differential equation that is, in physical terminology, the *continuity equation* of mass in a measure transport [126, 168, 44, 180]:

$$\partial_t\mu_t = \operatorname{div}(\nabla\delta\mathcal{E}(\mu_t)\cdot\mu_t). \tag{24}$$

This formalism allows for obtaining optimization guarantees in arbitrarily wide shallow networks [44, 167], despite the fact that the functional *is not convex for displacements*, along with generalization bounds without the dimensional curse [10]. However, these optimization guarantees are valid in the thermodynamic limit, also called the *mean field* limit, where the evolution of the measure begins at μ_0 and not in its empirical version $\mu^{(m)}$. In the same way that we have a law of large numbers (and also a central limit theorem), at initialization $\mathbb{R}^n$(see equation (21)) the trajectories starting at μ_0 and $\mu^{(m)}$, let us call them μ_t and $\mu_t^{(m)}$, can be bounded, under certain conditions, so that they preserve the weak convergence rate of $1/\sqrt{m}$ uniformly in time [43].

In sum, the Eulerian view of shallow networks allows for the identification of a robust mathematical structure, in addition to studying the optimization problem with tools from measure transport and statistical mechanics. Although the learning guarantees are still qualitative (the number of neurons required to guarantee optimality in the general case remains exponential in the ambient

dimension), these tools allow for optimism regarding a mathematical theory of shallow networks. Extensions to deeper networks have begun to be studied [154, 60], as well as problems with symmetries [223], or competitive optimization problems in zero-sum games [54].

5 Generalization

An learning algorithm has good generalization capacity if the hypothesis $h \in \mathcal{F}$ it chooses differs little from the expert that produced the examples; that is, if the loss $L(h)$ is small. How can this condition be guaranteed if the learning algorithm only knows the examples $\mathcal{D}$ and, in one way or another, the space $\mathcal{F}$? The problem may seem impossible considering that $\mathcal{D}$ is always finite, that the dimensional curse always lurks around every corner, and that $\mathcal{F}$ is, as a general rule, infinite.

If it has a solution, what form can we expect an statement about generalization capacity to take? Since it involves finding a bound for $L(h)$ as a function of what the learning algorithm knows, we can expect an expression for this bound involving the empirical risk of h, $\hat{L}_{\mathcal{D}}(h)$, the size of $\mathcal{D}$, n, and some measure of the capacity or richness of $\mathcal{F}$, which we can abstractly denote as $\mathsf{Cap}(\mathcal{F})$.

Furthermore, the statement must necessarily be probabilistic, which means the bound will be valid with a certain probability, expressed as a confidence $\geqslant 1-\delta$ for a small fixed value δ. In summary, following the usual path in statistical learning, we will have expressions of the form:

$$L(h) \leqslant_{\delta} \hat{L}_{\mathcal{D}}(h) + \mathsf{Fun}(\mathsf{Cap}(\mathcal{F}), n, \delta)$$

where $\leqslant_{\delta}$ means with probability at least $1-\delta$ relative to the samples $\mathcal{D}$, and where Fun is a certain function of the arguments $\mathsf{Cap}(\mathcal{F})$, n, and δ. The following result is an example to illustrate this procedure:

Theorem 4 ([176], Corollary 4.6)**.** *Suppose that $\mathcal{F}$ is a finite set of binary hypotheses. Then, for any $\delta > 0$,*

$$L(h) \leqslant \hat{L}_{\mathcal{D}}(h) + \sqrt{\frac{\ln(|\mathcal{F}|) + \ln(2/\delta)}{2n}}$$

Thus we can interpret that $\mathsf{Cap}(\mathcal{F}) = |\mathcal{F}|$. The result tells us that generalization increases if the minimization of the empirical risk ensures that $\hat{L}_{\mathcal{D}}(h)$ is small and that n is large enough so that the second term of the expression is also small.

An alternative is to express the relationship $L(h) \leqslant_{\delta} \mathsf{Fun}$ as a lower bound on n (*sample complexity*). For example, the previous theorem can be expressed by saying, under the same hypotheses, that $|L(h) - \hat{L}_{\mathcal{D}}(h)| \leqslant \epsilon$ if

$$n \geqslant \frac{1}{2\epsilon^2}(\ln|\mathcal{F}| + \ln\frac{2}{\delta}). \tag{25}$$

It should be noted that (25) and, therefore, Theorem 4, can be improved in the *realizable* case, that is, when $f^* \in \mathcal{F}$. Indeed, under these conditions, a linear bound in ϵ^{-1} can be established [176, Corollary 2.3]:

$$n \geqslant \frac{1}{\epsilon}(\ln|\mathcal{F}| + \ln\frac{1}{\delta}). \tag{26}$$

However, hypothesis spaces are usually infinite, and the study of the generalization that can be achieved is more sophisticated, as it is necessary to introduce appropriate notions of $\mathsf{Cap}(\mathcal{F})$ and establish upper bounds for $L(h)$ or lower bounds for n that ensure generalization with confidence $1-\delta$. In this section, we introduce several notions of the *capacity* of a hypothesis space $\mathcal{F}$. The Vapnik-Chervonenkis (VC) dimension, introduced as an *index* in [203, §1], is valid for binary hypotheses and does not depend on the probability density P over $\mathcal{X}$, a fact expressed by saying that it is an *agnostic* notion. There are good expositions of this in many texts, such as [4], [108] or [176], and we dedicate §5.1 to it. The second section, §5.2, is devoted to the so-called *Pollard dimension* (Pol), introduced in [158], which is similar to the VC dimension but without the restriction to binary values. The third section, §5.3, is dedicated to the so-called *Rademacher dimension* (Rad). This concept (explained in detail in [176, chapter 26], but see also the notes [12]) is not agnostic, and this fact, as we will see, usually allows for achieving better bounds on the quantities of interest. Finally, in §5.4 we record several comparisons between the three notions of capacity just mentioned.

5.1 The VC Dimension

Let us first look at the case of a space $\mathcal{F}$ of binary hypotheses $h : \mathcal{X} \to \{0,1\}$, which we can identify with subsets of $\mathcal{X}$: $h \leftrightarrow h^{-1}(1) = 1_{h(x)=1}$. We say that $\mathcal{F}$ *shatters* a finite set $Z \subset \mathcal{X}$ if every binary function on Z is the restriction of some $h \in \mathcal{F}$. The Vapnik-Chervonenkis *dimension* (or *capacity*) of $\mathcal{F}$, denoted $\mathsf{VC}(\mathcal{F})$, is defined as the maximum cardinality of a finite subset $Z \subset \mathcal{X}$ shattered by $\mathcal{F}$, if this maximum exists, or ∞ otherwise. Thus we see that VC is a purely combinatorial concept. In terms of practical calculation, for a finite capacity k one normally exhibits a set Z of cardinality k shattered by $\mathcal{F}$ and then proves that no subset of cardinality $k+1$ can be shattered. In the case of infinite capacity, it must be shown that for every $k \in \mathbb{N}$ there exists a set Z of cardinality k shattered by $\mathcal{F}$.

Examples

(1) Let $\mathcal{X} = \mathbb{R}$ and $\mathcal{F}$ be the set of positive half-lines $[a,\infty)$, $a \in \mathbb{R}$. Then $\mathsf{VC}(\mathcal{F}) = 1$. Indeed, given $Z = \{z\}$, $z \in \mathcal{X}$, there are half-lines that contain it and others that do not, so that Z is shattered by $\mathcal{F}$. On the other hand, if $Z = \{z, z'\}$, $z < z'$, a half-line containing z also contains z' and, therefore, Z cannot be shattered by $\mathcal{F}$.

(2) Still in $\mathbb{R}$, consider the set $\mathcal{F}$ of closed intervals. Then $\mathsf{VC}(\mathcal{F}) = 2$, since if $Z = \{z, z'\}$ as in the previous example, there are closed intervals disjoint from Z, others that contain it, and still others that contain z and not z', and vice versa. In contrast, if $Z = \{z, z', z''\}$, $z < z' < z''$, there is no closed interval that contains z and z'' without containing z', so no Z of cardinality 3 can be shattered by $\mathcal{F}$. Note that in example (1) we would also have capacity 2 if, in addition to positive half-lines, we simultaneously considered negative ones.

(3) Let $\mathcal{F}$ be the space of half-planes in the plane $\mathcal{X} = \mathbb{R}^2$. A set Z of three non-collinear points in $\mathcal{X}$ is shattered by $\mathcal{F}$, since for every subset Z' of Z there are half-planes h such that $h \cap Z = Z'$. On the other hand, no set of four distinct points in $\mathcal{X}$ can be shattered by $\mathcal{F}$. Indeed, we can assume that no three are collinear, as a half-plane containing the two ends of a segment contains any other point of that segment. We can also assume that no point is inside the triangle formed by the other three, as then this point automatically belongs to any half-plane containing the triangle. And if the four points form a convex quadrilateral, there is no half-plane that contains the ends of one diagonal and excludes the ends of the other diagonal. In summary, no set of four points can be shattered by $\mathcal{F}$, hence $\mathsf{VC}(\mathcal{F}) = 3$.

(4) The previous example is valid in any dimension $d \geqslant 2$: the VC capacity of the set $\mathcal{F}$ of half-spaces of $\mathbb{R}^d$ is $d + 1$. The easiest part is finding a set Z of $d + 1$ points shattered by $\mathcal{F}$. Let $Z = \{z_0, z_1, \ldots, z_d\}$, where z_0 is the origin and z_j is the unit point on the j-th coordinate axis. Let $\{0, 1, \ldots, d\} = A \sqcup B$ be an arbitrary partition and set $w_j = 1$ if $j \in A$ and $w_j = -1$ if $j \in B$. Then the half-space defined by the hyperplane with equation $h(x) = w_0/2 + w_1 x_1 + \cdots + w_d x_d = 0$ contains (excludes) the points z_j for $j \in A$ ($j \in B$), since $h(z_0) = w_0/2$ and $h(z_j) = w_0/2 + w_j$ for $j > 0$, from which it follows that $h(z_j)$ has the same sign as w_j for all j. The remaining part (that no set of points Z of cardinality $d+2$ can be shattered by $\mathcal{F}$) is a consequence of the fact that there exists (Radon's lemma) a partition $Z = Z' \sqcup Z''$ such that $[Z'] \cap [Z''] \neq \emptyset$, where we use $[Z']$ and $[Z'']$ to denote the convex hulls of Z' and Z'': if there were a half-space separating Z' and Z'', this half-space would also separate $[Z']$ and $[Z'']$.

(5) The VC capacity of the family of convex polygons with r or fewer sides is $2r+1$. Indeed, let Z be a set of $2r+1$ points on the circumference of a circle and choose any partition $Z = Z' \sqcup Z''$. If $|Z'| \leqslant r$, the convex polygon with vertices Z' is contained in the circle and, therefore, excludes the points of Z''. If, on the other hand, $|Z''| \leqslant r$, then we can construct, from the tangents to the circumference at the points of Z'', a convex polygon of $|Z''|$ sides that includes Z' and excludes Z'': it suffices to move the said tangents an infinitesimal amount toward the center. These arguments can be easily modified to establish that convex polygons of exactly r sides also

have capacity $2r+1$. Finally, it is clear that convex polygons, without conditions on the number of sides, have capacity ∞.

The role of VC in the generalization problem is made clear in the following theorem:

Theorem 5. *Consider $k = \mathsf{VC}(\mathcal{F})$ and suppose $k < \infty$. Then we have:*

(1) $L(h) \leqslant_\delta \hat{L}_{\mathcal{D}}(h) + O\left(\sqrt{\frac{k+\ln(1/\delta)}{n}}\right)$.

Alternatively, $n = O\left(\frac{1}{\epsilon^2}(k+\ln(1/\delta))\right)$ ensures $|L(h) - \hat{L}_{\mathcal{D}}(h)| \leqslant_\delta \epsilon$.

(2) *If $h \in \mathcal{F}$ satisfies $\hat{L}_{\mathcal{D}}(h) = 0$, then $n \geqslant \frac{8}{\epsilon}(k\ln(16/\epsilon) + \ln(2/\delta))$ guarantees $L(h) \leqslant_\delta \epsilon$.*

The expression for the lower bound of n is $O\left(\frac{k}{\epsilon}\ln(1/\epsilon) + \frac{1}{\epsilon}\ln(1/\delta)\right)$. In terms of the cost, $L(h) \leqslant_\delta O\left(\frac{1}{n}(k\ln(n/k) + \ln(1/\delta))\right)$ if $\hat{L}_{\mathcal{D}}(h) = 0$.

Finally, let's note what happens when VC is ∞ (cf. [176, theorem 6.6]):

Theorem 6. *For a class $\mathcal{F}$ of binary hypotheses such that $\mathsf{VC}(\mathcal{F}) = \infty$, there exist oracles that no algorithm can learn.*

5.2 The Pollard Dimension

Suppose $\mathcal{Y} = [0, K] \subset \mathbb{R}$, $K > 0$, so that $\mathcal{F}$ is a set of functions $h : \mathcal{X} \to [0, K]$. Let $X = \{x_1, \ldots, x_n\}$ be a subset of $\mathcal{X}$. We say that $\mathcal{F}$ *shatters X with witnesses* $t_1, \ldots, t_n \in \mathbb{R}$ if for every subset $X' \subset X$ there exists a function $h \in \mathcal{F}$ such that $h(x_j) \leqslant t_j$ or $h(x_j) > t_j$ depending on whether $x_j \in X'$ or $x_j \notin X'$.

The *Pollard dimension*, $\mathsf{Pol}(\mathcal{F})$, is the maximum cardinality of a subset X of $\mathcal{X}$ that $\mathcal{F}$ can shatter. In the binary case, $\mathsf{Pol}(\mathcal{F}) = \mathsf{VC}(\mathcal{F})$. Another example is $\mathsf{Pol}(\mathcal{F}) = \dim(\mathcal{F})$ if $\mathcal{F}$ is a finite-dimensional vector space of real functions.

With the previous notation, let $p = \mathsf{Pol}(\mathcal{F})$ and let $\mathcal{D} \sim P^n$, that is, a subset of n elements of $\mathcal{X}$ drawn independently according to the distribution P. Then we have:

Theorem 7 (Pollard, 1984,[158])**.** *For any $h \in \mathcal{F}$:*

$$|\hat{L}_{\mathcal{D}}(h) - \mathbb{E}_{x\sim P}[h(x)]| \leqslant_\delta K\sqrt{\frac{2p}{n}\ln\frac{en}{p}} + K\sqrt{\frac{1}{2n}\ln\frac{1}{\delta}}.$$

Alternatively, $|\hat{L}_{\mathcal{D}}(h) - \mathbb{E}_{x\sim P}[h(x)]| \leqslant_\delta \epsilon$ for $n \geqslant \frac{8K^2}{\epsilon^2}\left(p\ln\frac{8K^2}{\epsilon^2} + \frac{1}{4}\ln\frac{1}{\delta}\right)$.

5.3 Rademacher Complexity

This notion can be defined for a family $\tilde{\mathcal{F}}$ of functions $\tilde{h} : Z \to [a, b] \subset \mathbb{R}$, where (Z, P) is a probability space, and its purpose is to calibrate the capacity of $\tilde{\mathcal{F}}$ to fit a certain random noise. In the application to the basic model, we will

have $z = (x, y)$, with $P(z) = P(x, y) = P(y|x)P(x) = P_x(y)P(x)$, and $\tilde{\mathcal{F}}$ will be the family of functions $\tilde{h} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, $h \in \mathcal{F}$, defined by the relation (the pointwise defect ℓ was introduced in the paragraph **Loss** in §2.2, page 13):

$$\tilde{h}(x, y) = l(h(x), y). \tag{27}$$

In reality, we must consider two notions of *Rademacher complexity*: $\mathsf{Rad}_{\boldsymbol{z}}(\tilde{\mathcal{F}})$, the *empirical Rademacher complexity* relative to a sample $\boldsymbol{z} = \{z_1, \ldots, z_n\}$, and $\mathsf{Rad}_n(\tilde{\mathcal{F}})$, the complexity for samples of length n. To define them, we introduce *Rademacher variables* $\boldsymbol{\sigma} = \sigma_1, \ldots, \sigma_n$ to represent random binary noise. These are independent random variables, one for each data point in the sample, taking values in $\{-1, 1\}$ with equal probability.

The correlation of a sample of this noise with the values $\tilde{h}(\boldsymbol{z}) = \tilde{h}(z_1), ..., \tilde{h}(z_n)$ is expressed by the scalar product $\boldsymbol{\sigma} \cdot \tilde{h}(\boldsymbol{z})$. Therefore, $\sup_{\tilde{h} \in \tilde{\mathcal{F}}} \boldsymbol{\sigma} \cdot \tilde{h}(\boldsymbol{z})/n$ expresses the best correlation that can be obtained, on average over the sample, with functions from $\tilde{\mathcal{F}}$. With all this, we can now define the two Rademacher complexities:

$$\mathsf{Rad}_{\boldsymbol{z}}(\tilde{\mathcal{F}}) = \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\tilde{h} \in \tilde{\mathcal{F}}} \frac{\boldsymbol{\sigma} \cdot \tilde{h}(\boldsymbol{z})}{n} \right] \quad \text{and} \quad \mathsf{Rad}_n(\tilde{\mathcal{F}}) = \mathbb{E}_{\boldsymbol{z} \sim P^n}[\mathsf{Rad}_{\boldsymbol{z}}(\tilde{\mathcal{F}})]. \tag{28}$$

Theorem 8. *For all $\delta > 0$ and all $\tilde{h} \in \tilde{\mathcal{F}}$, the following inequalities hold:*

$$\mathbb{E}_{z \sim P}[\tilde{h}(z)] \leqslant_\delta \hat{L}_{\boldsymbol{z}}(\tilde{h}) + 2\mathsf{Rad}_{\boldsymbol{z}}(\tilde{\mathcal{F}}) + 3\sqrt{\frac{\ln \frac{2}{\delta}}{2n}} \tag{29}$$

$$\mathbb{E}_{z \sim P}[\tilde{h}(z)] \leqslant_\delta \hat{L}_{\boldsymbol{z}}(\tilde{h}) + 2\mathsf{Rad}_n(\tilde{\mathcal{F}}) + \sqrt{\frac{\ln \frac{1}{\delta}}{2n}} \tag{30}$$

If we specialize the previous results to the case of the basic model, via the introduction of the space $\tilde{\mathcal{F}}$ associated with $\mathcal{F}$ explained at the beginning of this section, we obtain:

Theorem 9 (Rademacher bound)**.** *Let $\mathcal{F}$ be a hypothesis space and $\mathcal{D}$ a set of empirical data. Then for all $h \in \mathcal{F}$, the following inequality holds:*

$$L(h) \leqslant_\delta \hat{L}_{\mathcal{D}}(h) + 2\mathsf{Rad}_n(\mathcal{F}) + \sqrt{\frac{\ln(1/\delta)}{2n}} \tag{31}$$

If we abstract the role of $\tilde{h}(\boldsymbol{z}) \in \mathbb{R}^n$ by viewing it merely as a vector $\boldsymbol{a} \in \mathbb{R}^n$, we can define $\mathsf{Rad}(A) = \mathbb{E}_{\boldsymbol{\sigma}}[\sup_{\boldsymbol{a} \in A} \frac{\boldsymbol{\sigma} \cdot \boldsymbol{a}}{n}]$ for every subset A of $\mathbb{R}^n$, a notion that facilitates the establishment of properties of Rad.

Theorem 10 (Bounding $\mathsf{Rad}(A)$)**.** *Let $A \subset \mathbb{R}^n$ be finite and define $L = \sup_{\boldsymbol{a} \in A} \|\boldsymbol{a} - \bar{\boldsymbol{a}}\|$, where $\bar{\boldsymbol{a}}$ is the barycenter of A. Then:*

$$\mathsf{Rad}(A) \leqslant \frac{L\sqrt{2 \ln |A|}}{n} \tag{32}$$

If $\mathcal{F}$ is a class of binary hypotheses such that $\mathsf{VC}(\mathcal{F}) = k$, and $\mathcal{D} \sim P^n$, then:

$$\mathsf{Rad}_{\mathcal{D}}(\mathcal{F}) \leqslant \sqrt{\frac{2k \ln n}{n}} \tag{33}$$

It is worth noting here that finding good bounds for Rademacher complexity in neural networks is currently an active research front.

5.4 Ramifications

We first take a look at bounds through other avenues; then, at non-uniform bounds; and finally, at an issue we consider especially important: the phenomenon called *double descent* that manifests itself in the training of deep neural networks and that refutes the conventional understanding of the trade-off between training error and generalization error (Figure 1).

Bounds by other means. Generalization bounds can be obtained by other means, such as using notions of algorithmic stability [27]. These notions are based on *sensitivity analysis*, which seeks to quantify the impact on a system's output of a variation in the input with the purpose of being able to design resilient systems against perturbing input noise. For the state of the art on the bounds obtained with these approaches, we have the recent article [28], which additionally obtains lower bounds of the generalization error and substantially better upper bounds than those previously known.

Non-uniform bounds. The dimensions VC (agnostic regarding the distribution P that governs data generation) and Rad (which depends on P) have a uniform character in the ball $\mathcal{F}_\delta$, a condition that entails certain limitations, [134], and which corresponds to the *statistical* rate $O(1/\sqrt{n})$ of generalization.

In more favorable situations, it is possible to replace this uniform analysis with a more refined analysis based on the local complexity of the hypothesis space around the solution of the regularized ERM, culminating in a *fast rate* $O(1/n)$ in the case of certain Gaussian kernel spaces [16]. This refined analysis can also be applied in sparse regression models [208].

The double descent paradox. Another "pessimistic" aspect of the generalization bounds seen so far is that they are agnostic to the optimization algorithm used. The importance of also considering this aspect is illustrated by the "double descent" paradox. Figure 9 outlines a phenomenon that exemplifies the contrast between the conventional understanding of the so-called *trade-off between hypothesis space complexity and learning rate* (bias-variance trade-off) [176, chapter 5] and what recent years of experimentation with deep neural networks have revealed [134, 23, 22].

The trade-off in question seeks to balance underfitting and overfitting by choosing a model that is expressive enough to extract the relevant structure of the data, but also simple enough not to pick up spurious details. According to this perception, neural networks with many neurons (or *overparameterized*)

would be discarded, since the high number of parameters they depend on allows them to memorize all the training data and, because of this, their generalization capacity should be very low or null. However, practice shows that they can indeed fit the training data exactly and at the same time have a generalization capacity that, in fact, increases with the complexity of the deep network. Fortunately, the core of the paradox has been easily explained in [126] using random matrix theory, a remarkable article in which it is also shown that the optimization algorithm plays a fundamental role in the *implicit* regularization of the ERM, which also explains (for tractable models such as generalized linear models) the good generalization of overparameterized neural networks.

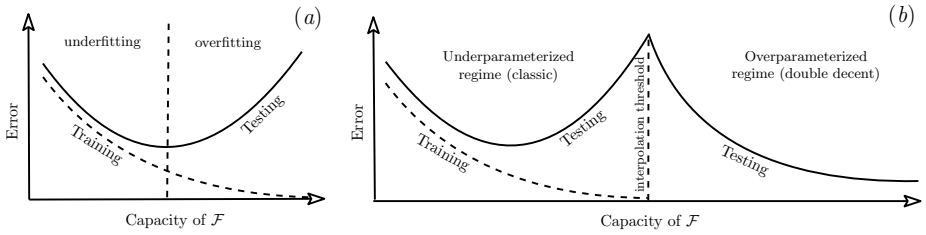

Figure 9: Adaptation of Figure 1 from [22], also taking into account the graphs from [126]. (a) This graph is similar to that of Figure 1 but here the horizontal axis shows the capacity of $\mathcal{F}$. It represents a description that governs when the number of parameters of the functions in $\mathcal{F}$ is less than the number of data points available in the training: as capacity increases, the training error decreases monotonically, but the generalization error, measured using test data, first decreases and from a certain value, which separates underfitting from overfitting, increases. (b) Represents the extraordinary behavior, and paradoxical at a first glance, that appears when the number of parameters exceeds the number of training data points: the generalization error decreases again from the peak that separates the two regimes, which enables practically perfect learning of the training data and at the same time a generalization capacity that increases with the number of parameters.

To conclude this section, it is befitting to quote Donoho [55] regarding the "blessings of dimensionality', which "are less widely perceived, but include the phenomenon of measure concentration (so-called in the geometry of Banach spaces), which means that certain random fluctuations are very well controlled in high dimensions, and also the success of asymptotic methods, widely used in mathematical statistics and statistical physics, which suggest that statements can be made in very high-dimensional settings that happen to be too complicated in moderate dimensions".

6 Other models and open problems

We have been seeing that ML has connections with various domains, particularly mathematics and statistics, but also algorithmics and complexity, mathematical physics, or combinatorics. This underlines its transversal and multidisciplinary

character, but this list can be extended with other domains that can offer important opportunities for research, whether theoretical or applied.

In the next section, §6.1, some of the connections that seem most relevant to us are briefly indicated. The following two sections are devoted to a slightly more in-depth consideration of the relationship with the general problem of incorporating explainability into ML, §6.2, and with the area of algebro-geometric NNs, §6.3. In the last section, §6.4, some open questions or issues that should deserve more attention are indicated.

6.1 Connections

The selection of this section moves away from mere curiosity and instead focuses on lines that provide a fuller understanding of ML and are relevant for future inquiries.

Information Theory. Perhaps the treatise [119] is what best represents this connection, since its answer to the question of why it is beneficial to unify information theory and ML is that "they are two sides of the same coin", and adds: "Brains are the ultimate compression and communication systems, and the state-of-the-art algorithms for data compression and error-correcting codes use the same tools as machine learning".

Of the more specific aspects, we highlight the contributions of N. Tishby and collaborators in what they call the "information bottleneck method/principle". Introduced in [193], it is applied to deep ML in [194] and [177]. The most recent work we have found in this regard, [157], is in collaboration with Z. Piran and R. Shwartz-Ziv, and its object is to introduce what they call a "dual bottleneck".

ML with mathematically generated data. The work [111] is paradigmatic of this connection and suggests previously unsuspected possibilities. One of the ideas found there is to randomly generate a large sample of mathematical expressions f, in an appropriate format, and form the list of pairs (f', f), where f' is the derivative of f (relatively easy to calculate using well-known algorithms). The learning algorithm must predict f from f'. The authors explain what they have discovered thus: "Neural networks have a reputation for being better at solving statistical or approximate problems than at performing calculations or working with symbolic data. In this paper, we show that they can be surprisingly good at more elaborated tasks in mathematics, such as symbolic integration and solving differential equations. We propose a syntax for representing mathematical problems, and methods for generating large datasets that can be used to train sequence-to-sequence models. We achieve *results that outperform commercial Computer Algebra Systems such as Matlab or Mathematica*". The most surprising thing is that the learning algorithm does not know any theory of integration nor any of the rules we learn in the first courses of analysis.

We also highlight some works by Yang-Hui He and collaborators that obtain analogous results in other domains: [81] (a monograph on questions raised by string theory, and especially by algebraic and enumerative geometry); [82]

(algebraic structures); [2] (number theory and the Birch-Swinnerton-Dyer conjecture); and [83] (aimed at finding out the behavior of AL in questions related to the study of finite graphs, the authors write that "NNs can perform, with high efficiency and precision, a multitude of tasks ranging from the recognition of the Ricci flatness condition of a graph, to the prediction of the spectral gap or the detection of the presence of Hamiltonian cycles").

Also fitting perfectly at this point are the works [48] and [110]. The latter considers NNs on graphs and offers a perspective on how they can be treated with ML.

AL in Physics. Physicists are also interested in the impact of ML on their work. Some, such as [214], seek an "artificial physicist" that can learn without supervision. In a similar line we find [94].

In contrast, [41] offers an extensive overview of the connections between the physical sciences and AL. In this sense, it is appropriate to take into account the initiative [101] of the United States Department of Energy to create a quantum version of the Internet, which we see as a further stimulus to continue investigating AL.

6.2 Causality and Explainable ML

Concerning these topics, the ideas of Judea Pearl must be placed at the forefront, as explained in detail in works such as the book [145] (which in particular contains the remarkable belief propagation algorithm in graphs), the lecture [146] (art and science of causes and effects), and [147] (the mature fruits of the author's research for more than two decades, a treatise on the foundations of causality). As a complementary text, that appeared after the first edition of [147], see [181] (remarkable for its clarity and conci,...,seness) and [68] (a brief article analyzing the area from the point of view of the social sciences: "All becomes more difficult when we shift our focus from *What* to *What-if* and *Why*").

We continue with J. Pearl and some fundamental works appeared in the last five years: [148] (causes of effects and effects of causes); [149] (an appreciation of Trygve Haavelmo as one of the initiators of causal calculations); [199] ("In contrast to textbook approaches such as expectation maximization and the gradient method, our approach is non-iterative, yields closed form parameter estimates, and eliminates the need for inference in a Bayesian network"); and [14] (on the problems involved in data fusion in relation to causality). Finally, two quite different but enlightening pieces: [150] (discusses seven obstacles to ML from the point of view of causality) and [151] (a relatively informal presentation of his mature thought). We add the text [153] (introduction to causal inference and foundations of ML), which the large number of citations of Pearl's works confers a character of recognition to this researcher.

Explainability. On this topic, closely related to causality and interpretability, we select a few recent works that treat it from various points of view and that seem to us meritorious sources to begin its study: [59, 218, 8, 7, 66].

6.3 Algebro-geometric Neural Networks

The quantities x and w of the neuron model we considered in §2.3 are real numbers. But we can imagine them to be entities of an algebraic structure $\mathcal{A}$ sufficient to ensure that the expression $x \cdot w = x_1 w_1 + \cdots + x_n w_n$ and an activation function $\sigma : \mathcal{A} \to \mathcal{A}$ make sense. For example, $\mathcal{A}$ can be a finite-dimensional real algebra and σ the the result of applying an ordinary sigmoid acting component-wise (with respect to a prefixed base of the algebra). We thus arrive at the concept of $\mathcal{A}$-*neuron* and, connecting neurons as we did in the mentioned section, at the notion of $\mathcal{A}$-*neural network*, or $\mathcal{A}$-NN. Another generalization consists in replacing x and w with more general data structures, such as $\mathcal{A}$-tensors (or $\mathcal{A}$-arrays), and the product $x \cdot w$ with an appropriate operation $x * w$. Among these operations, the most used are certain bilinear products, such as *convolution*, and also non-linear ones, such as *max-pooling*.

Thus, the usual neurons and neural networks are $\mathbb{R}$-neurons and $\mathbb{R}$-neural networks. Beyond real numbers, among the most immediate concrete cases of algebras $\mathcal{A}$ we can mention $\mathbb{C}$ (complex numbers), $\mathbb{H}$ (quaternions), $\mathbb{O}$ (octonions), a matrix algebra $\mathbb{R}(n)$, or a geometric algebra $\mathcal{G} = \mathcal{G}_{r,s}$ of signature (r, s). To simplify the terminology, we will speak of real, complex, quaternionic (QNN), octonionic, matricial, and geometric (GNN) networks, respectively.

The reason for using geometric algebras is that their formalism is optimally adapted to expressing the geometric facts of any *linear geometric space*, that is, a real vector space $E = E_{r,s}$ endowed with a metric of signature (r, s). The most direct way to introduce the geometric algebra $\mathcal{G}_{r,s}$ of this space, and at the same time closest to the ideas on which W. K. Clifford (1845–1879) based its creation, is that the exterior algebra $\Lambda E_{r,s}$ admits an **associative** bilinear product (baptized as *geometric product* by Clifford himself) such that

$$xx' = x \cdot x' + x \wedge x', \quad x, x' \in E, \tag{34}$$

that is, it amalgams the two products (inner and outer) introduced by Grassmann (see [216] for a detailed presentation of this approach, or [215, chapter 3] for an axiomatic presentation). Thus we can think of $\mathcal{G}_{r,s}$ as the exterior algebra enriched with the geometric product (this structure is also known as *Clifford algebra*). It is clear that it has dimension 2^n, where $n = r + s = \dim E$.

Note that equation (34) shows that the linear grading of $\mathcal{G}_{r,s}$, which is in fact a grading with respect to the outer product, is not so with respect to the geometric product. But the decomposition $\mathcal{G} = \mathcal{G}^+ \oplus \mathcal{G}^-$ into components of even degrees ($\mathcal{G}^+$) and odd degrees ($\mathcal{G}^-$) is a grading modulo 2 *also with respect to the geometric product* (ultimately this derives from equation (34), in which the product of two vectors is the sum of a scalar, which has grade 0, and a bivector, which has grade 2). In particular, $\mathcal{G}^+$ is a subalgebra. The isomorphisms $\mathcal{G}_{1,0} \simeq \mathbb{R} \oplus \mathbb{R}$, $\mathcal{G}_{0,1} \simeq \mathcal{G}^+_{2,0} \simeq \mathbb{C}$, $\mathcal{G}_{2,0} \simeq \mathbb{R}(2)$ or $\mathcal{G}_{0,2} \simeq \mathcal{G}^+_{3,0} \simeq \mathbb{H}$, easy to establish directly, are in fact examples of a general pattern (cf. [215]). Of these isomorphisms, those that most closely connect algebra with geometry are $\mathbf{C} = \mathcal{G}^+_{2,0} \simeq \mathbb{C}$ and $\mathbf{H} = \mathcal{G}^+_{3,0} \simeq \mathbb{H}$ in the case of Euclidean plane and space (we will say that $\mathbf{C}$ and $\mathbf{H}$ are the *geometric* complex numbers and quaternions,

respectively, as they emerge directly from geometry, not from an *ad hoc* algebraic definition like the usual ones invoked to introduce $\mathbb{C}$ and $\mathbb{H}$). For samples of various applications of $\mathcal{G}$, see [215, 112] and their bibliographies.

Next we provide some general references on various algebro-geometric NNs together with some brief comments. For more detailed considerations on these and other references, see [217].

Complex neural networks. Perhaps the most important idea of these networks is that they can exploit the phase properties of complex numbers. At the beginning of the study of these networks, the works of Hirose and his school stand out, very focused on signal processing, with collections such as [85] (2003) and treatises like [141] (2009) or [86] (2012; a second edition of a book of the same title and author was published in 2006), and the collection of ten articles gathered in [87] (2013), of which the first one stands out, by Hirose himself (the editor of the volume), titled *Application fields and fundamental merits of complex-valued neural networks*. To the same circle belongs the text [1], which illustrates with very convincing graphic experiments the value of taking the phase into account.

More recently we have [75] (2016), on complex convolutional networks; [159] (2017), focused on image classification; [196] (2017), where the emphasis lies on deep networks; and [129], which provides an assessment of complex networks in tasks of classification of real signals.

Finally, we mention [198], in which the significance of complex networks from other points of view is elucidated, particularly in the perspective of *geometric deep learning*, seen, as stated in [31] as "umbrella term for emerging techniques that attempt to generalize deep (structured) neural models to non-Euclidean domains such as graphs and manifolds."

Quaternionic networks. The interest of these networks stems from the relationship that (geometric) quaternions have with the rotation group of the ordinary Euclidean space, a relationship that is especially transparent in terms of $\mathbf{H}$, since the expression $\underline{h}(x) = hxh^{-1}$, $h \in \mathbf{H}$ not zero, is a vector if x is a vector and $\underline{h}$ is a similarity of dilation ratio $|h|^2$ (hence a rotation if h is unitary). Another reason is that quaternions have three phases and these phases can be used to extract valuable information from the signals to be processed.

Research on QNNs also goes back quite a long way, even before that of complex networks. We refer to [29] and [88] for relevant historical information regarding what is called *Clifford analysis*, especially in relation to Fourier and wavelet transforms in a quaternionic context and in its generalization to the geometric context. In the more specific topic that concerns us, at the origin we find Gerald Sommer and his collaborators: [38] (1998, generalization of Gabor filters) and [36] (2000, generalization of the multilayer perceptron). The report [42] presents a theory of quaternionic wavelets "for image analysis and processing" and [92] (2009) offers an overview of the properties and applications of quaternionic networks up to that moment.

In the last decade, research on QNNs has proceeded on both the applied

and theoretical fronts. Article [93] deals with the quaternionic multilayer perceptron. In [103], Hopfield QNNs and their rotation invariance are investigated. In the works [142] and [143], QNNs are applied to the understanding of spoken language. Deep QNNs are studied in [67] and the convolutional ones in [222]. Finally, [219] presents a quaternionic version of capsule networks (aimed at processing point clouds of the Euclidean space) and in [130] a QNN is introduced that provides contrast invariance and sensitivity to rotation angles.

Geometric networks Curiously, the study of GNNs began even before that of QNNs, as for example in [152], and G. Sommer was a promoter at the beginning of the millennium with works like: [182], in which he develops the theoretical foundations used for problems such as computer vision and robotics; [37], dedicated to a $\mathcal{G}$-version of the multilayer perceptron; and [61, 39], which develop the notion of the *monogenic signal*. A culmination of these efforts is the thesis of Sven Buchholz, [35], which should be considered, as its title indicates, a theory of neural computation with geometric algebras. As a sample of applications, we cite: [162] (image segmentation); [20] (support vectors in the geometric context); the volumes [18] (geometric computation for wavelet transforms, computer vision, learning, control, and action); and [21] (geometric computation in engineering and informatics); [63] (use of geometric algebra for edge detection in color images); [99] (optimization techniques in geometric estimation); [155] (clustering methods based on conformal geometric algebra $\mathcal{G}_{4,1}$); and [209] (multispectral image processing with geometric algebra). We end with [19], the first volume of what is intended to be a systematic treatise of these developments.

Other Algebro-geometric Networks. In the article [213] (2020), quaternionic convolutional networks are constructed and applied to CIFAR-10 and CIFAR-100 image classification. According to the authors, they have better convergence and accuracy than other networks applied to the same tasks. Octonions have also been applied with success to dictionary learning, as in [114] (2018), an approach that can in fact be formulated for more general algebras, including geometric ones, as explained in [115].

We now examine [89] (2020). In the summary it is stated that:

> ... our results demonstrate that there are alternative algebras which deliver better parameter and computational efficiency compared with $\mathbb{R}$. We consider $\mathbb{C}$, $\mathbb{H}$, $\mathbb{R}(2)$ (the set of 2×2 real-valued matrices), $\mathbb{C}(2)$, $\mathbb{R}(3)$, $\mathbb{R}(4)$, $\mathbb{R} \oplus \mathbb{R}$ (dual numbers) and the $\mathbb{R}^3$ with the cross product. Additionally, we note that multiplication in these algebras has higher compute density than real multiplication, a useful property in situations with inherently limited parameter reuse such as auto-regressive inference and sparse neural networks. We therefore investigate how to induce sparsity within AlgebraNets. We hope that our strong results on large-scale, practical benchmarks will spur further exploration of these unconventional architectures which challenge the default choice of using real numbers for neural network weights and activations.

Since this may not be the place for a critical appraisal, we limit ourselves to underlining what we have already commented on the geometric nature of $\mathbb{C}$,

$\mathbb{H}$, and $\mathbb{R}(2)$. Regarding the other cases, $\mathbb{R} \oplus \mathbb{R} \simeq \mathcal{G}_{1,0}$, $\mathbb{C}(2) \simeq \mathcal{G}_{1,3}$ (geometric algebra of spacetime), and $\mathbb{R}(4) \simeq \mathcal{G}_{2,2}$. The algebra $(\mathbb{R}^3, \times)$ is a Lie algebra, but in fact its nature is geometric, since the vector product of two vectors is the Hodge dual (within the algebra $\mathcal{G}_{3,0}$) of their exterior product. For all these matters you can consult [215]. As a counterpoint, we believe that the value of geometric networks is reinforced, both for theoretical reasons and for computational aspects.

6.4 Open Questions

The theory of AL is a field of active research, where particularly diverse mathematical aspects converge: information theory, to refine notions of complexity adapted to ultra-parameterized networks; statistical physics, to analyze computational and statistical limits in high dimensions; or harmonic analysis and optimization, crucial elements for describing—and prescribing—algorithms in deep networks. Consequently, there are currently a large number of open questions on all these fronts. Let us consider some of the most remarkable ones.

Deep functional models. As mentioned in section 3.2, deep neural networks pose an important challenge in terms of approximation and optimization. Mean-field methods give a satisfactory answer in the case of shallow networks (§4.3), and currently an important challenge is to be able to extend them to deep models, with preliminary results that can be found in [60].

Quantitative analysis of shallow networks. Shallow networks, despite being the simplest class of neural networks, are the subject of many open questions. In particular, recent results [71, 53] study lower bounds of learning; that is, they give negative results on the impossibility of learning certain classes of functions parameterized by shallow networks with a polynomial amount of data. In contrast, mean-field theory paints a more optimistic situation (and more aligned with empirical behavior), but for now in asymptotic terms. A problem of maximum importance is, therefore, understanding how these two perspectives can be linked with a quantitative version of the positive learning results.

Variations of the gradient descent method. The empirical results of deep networks depend critically on the optimization method. Although the gradient descent method is the most basic version and offers a clearer mathematical vision [184, 57, 44, 56, 30, 71], it is necessary to incorporate important variations, such as the stochastic version [169, 98, 46, 189, 97], or renormalization techniques [batch/layer normalization] [125, 168].

Beyond supervised learning. The landscape of open questions is encouraging. To mention one field in particular, self-supervised learning techniques are beginning to bear fruits comparable to, or even superior to, standard supervised versions [74]. Statistical learning theory is just beginning to consider learning regimes beyond the supervised one; for example, *transfer learning* [77], or the

aforementioned self-supervised learning [117, 195, 210], or the perspective of causality [6].

A Probability

In this appendix, we collect some concepts and results of probability theory used in various places in the preceding sections.

A.1 Hoeffding's Inequality

Consider $X_j \in [a,b]$, $j = 1,\dots,m$ independent random variables with the same mean μ, and define $\mu_m = \frac{1}{m}\sum_j X_j$. Then:

$$P(|\mu - \mu_m| > \epsilon) \leqslant 2e^{-2m\epsilon^2/(b-a)^2}.$$

As an example, we can consider the case $X_i \in [0, 2\pi]$ with a uniform distribution, so that $\mu = \pi$, $b - a = 2\pi$, and the exponent is equal to $-\frac{1}{2\pi^2}m\epsilon^2$; that is, $P(|\pi - \mu_m| > \epsilon) \leqslant 2e^{-\frac{1}{2\pi^2}m\epsilon^2}$.

A.2 Reciprocal Probability

Suppose that a random variable X satisfies $P(X > \epsilon) \leqslant f(\epsilon)$ for all $\epsilon > 0$, where $f(\epsilon) > 0$ is a given function. If $\delta' > 0$ is in the domain of f, and if $f(\delta') = \delta$, then $P(X \leqslant \delta') = 1 - P(X > \delta') \geqslant 1 - f(\delta') = 1 - \delta$. That is, we ensure the relationship $X \leqslant \delta'$ with a probability of at least $1 - \delta$.

Determining δ' as a function of δ is a matter that must be treated in each specific case. For example, in Hoeffding's inequality for uniform $X_i \in [0, 2\pi]$, $\delta' = \pi\sqrt{\frac{2}{m}\ln\frac{2}{\delta}}$. Thus, $|\pi - \mu_m| \leqslant \pi\sqrt{\frac{2}{m}\ln\frac{2}{\delta}}$ with a probability of at least $1 - \delta$. If we consider $Y_j = \cos X_j \in [-1, 1]$, which have mean 0, what we find is $|\frac{1}{m}\sum_j Y_j| \leqslant \sqrt{\frac{2}{m}\ln\frac{2}{\delta}}$ with a probability of at least $1 - \delta$.

A.3 McDiarmid's Inequality

Let $f : \mathcal{Z}^n \to \mathbb{R}$ be a function for which there exists a constant $c > 0$ such that:

$$|f(z_1,\dots,z_j,\dots,z_n) - f(z_1,\dots,z'_j,\dots,z_n)| \leqslant c \tag{35}$$

for all $j = 1,\dots,n$ and any $z_1,\dots,z_n,z'_j \in \mathcal{Z}$. If $\boldsymbol{z} = z_1,\dots,z_n \in \mathcal{Z}^n$ is a sequence of independent random variables, then the following inequalities hold:

$$P(f(\boldsymbol{z}) - \mathbb{E}[f(\boldsymbol{z})] \geqslant \epsilon) \leqslant \exp\left(\frac{-2\epsilon^2}{mc^2}\right) \tag{36}$$

$$P(f(\boldsymbol{z}) - \mathbb{E}[f(\boldsymbol{z})] \leqslant -\epsilon) \leqslant \exp\left(\frac{-2\epsilon^2}{mc^2}\right) \tag{37}$$

Using reciprocal probability, we can ensure that:

$$|f(\boldsymbol{z}) - \mathbb{E}[f(\boldsymbol{z})]| \leqslant c\sqrt{\frac{m}{2}\ln\frac{2}{\delta}} \tag{38}$$

with a probability not less than $1-\delta$. This inequality, called the "bounded differences" inequality, is a main ingredient in the proof of Rademacher inequalities.

A.4 Gibbs' Lemma and KL Divergence

Suppose that $p = p_1, \ldots, p_n$ and $q = q_1, \ldots, q_n$ are probability distributions. Then:

$$-\sum_j p_j \log_2(p_j) \leqslant -\sum_j p_j \log_2(q_j),$$

with equality if, and only if, $q = p$. It follows that $\sum_j p_j \log_2(p_j/q_j) \geqslant 0$, with equality only if $q = p$. This expression is usually known as the *Kullback-Leibler divergence* of p and q, and is denoted by $\mathsf{KL}(p, q)$. It should be noted that, in general, $\mathsf{KL}(q, p) \neq \mathsf{KL}(p, q)$.

The KL divergence is an important tool in the theory of Bayesian networks for comparing probability distributions at consecutive times. On the other hand, since $H(p) = -\sum_j p_j \log_2(p_j)$ is the *entropy* of the distribution p, i.e., the average information provided in a trial of p, it is natural that KL divergence is also significant in information theory.

B Bibliographical Notes

The bibliography on AL, not to say on AI, is vast and grows daily. Therefore it is impossible to give an account of it in a panoramic article like this one. Consequently, we limit ourselves to recording the bibliography that seems most relevant to us for each section, often with some comment or citation, and simultaneously suggesting various materials that can serve to delve deeper into the topics treated or into others of a complementary nature.

B.0 Notes to the Introduction

ML Treatises. General: [80, 176, 72, 161, 3]. Bayesian Approach: [192, 144, 172]. Applications: [173, 11]. Computational Aspects: [76, 135, 139].

Classic Articles: [123] (first mathematical model of biological neurons); [165] (introduction of the perceptron); [211] (ADALINE, adaptive linear neuron); [166] (neurodynamics); [171] (backpropagation algorithm); [127] (treatise on the perceptron, introduction to computational geometry, history of learning ideas and NNs up to 1988); [65] and [116] (visual pattern recognition with NN); [102] (collection of articles on perception as Bayesian inference); [133] (encyclopedia of cognitive sciences).

The Many Faces of AI: It is manifest that AI, via its accelerated development in all sectors, already affects all of humanity, and the direction and pace of its expansive evolution does not seem easy to guess. However, as indicated in [66], "society remains mostly ignorant of the capabilities and standard practices of AI today" while it "becomes aware of the dangers that ignorance entails and, rightly, demands solutions". The reflections, bibliography, and proposals of this article are valuable for everyone concerned by these uncertainties and by the incidence they have on the ethical sphere.

Among the documents that analyze the multiple facets of AI, current and future, the article [52] and report [188] seem very relevant to us. Writings of this kind, and like the one cited in the previous paragraph, are essential to form an overall idea, with very little technical apparatus, of AI as it is currently understood.

B.1 Notes to Section 1 (Preludes)

The treatise [132] offers a systematic presentation of AL from a probabilistic perspective. For a brief general appreciation of this perspective, see article [69]. The text [124] explains the historical evolution of the Bayes-Laplace formula from its origin to the publication of the book. It documents many applications, such as the fact that it was Alan Turing, in his work to decipher the Enigma, who promoted the systematic use of Bayes-Laplace theory in the form in which it has been known subsequently. A good complement is [186], written by one of the protagonists of the search for the atomic bomb lost in 1966 in the sea off the coast of Palomares (Spain) due to an American air force accident and also for the nuclear submarine *Scorpion* lost by the American navy in the Atlantic in May 1968.

With the subtitle "Why so many predictions fail–but some don't," the author of [179], perhaps the most prestigious haruspex in the prediction of electoral results, tells us that "Bayes's theorem [...] implies that *we must think differently about our ideas—and how to test them.* We must become more comfortable with *probability* and *uncertainty.* We must think more carefully about the assumptions and beliefs that we bring to a problem" (page 15, our italics).

The motto of the creator of modern pattern theory, Ulf Grenander (1923-2016), was to *use pattern theory to create mathematical structures derived from both the natural world and the world created by humans.* In this theory, Bayesian methods play a fundamental role, as reflected in [131]. In any case, pattern theory is much more general than the problem of pattern recognition, which has been the most studied until now, such as in [25].

Principal component analysis and singular value decomposition: [80, chapter 8], [113, chapter 7], [189], [208, chapter 8]. These methods are at the origin of "dimensionality reduction" techniques, an important aspect of ML [9]. A related topic is the so-called *discriminant analysis*, which can be found in most general treatises, such as [109, chapter 8].

Finally, [164] is a good reference for k-NN and k-means, which also includes computational treatments. We coincide with the author in the notion that

unsupervised learning is a very active area of research to which efforts must still be dedicated to obtain satisfactory general results.

B.2 Notes to section 2 (Inductive learning in high dimension)

Regarding the "curse of dimensionality," Donoho gave a precise description of it in [55]: "the apparent intractability of systematically searching through a high-dimensional space, the apparent intractability of accurately approximating a general high-dimensional function, the apparent intractability of integrating a high-dimensional function" (from the Abstract):

Of inductive learning, also called *statistical learning*, [201] contains an presentation of the structure of this theory by one of the initiators of its modern form. For a summary, see [202]. We also mention [108], a relatively elementary text, the extensive treatises [78] and [183], and the monograph [200].

In the presentation of inductive learning, we have resorted to the notion of an expert as a function $f^* : \mathcal{X} \to \mathcal{Y}$. In practice, an expert is also subject to a certain degree of uncertainty. The treatment of this situation can be done by replacing f^* with a probability distribution $P(x, y)$ over $\mathcal{X} \times \mathcal{Y}$, which we can view, via the relationship $P(x, y) = P(x)P(y|x)$, as a probability distribution $P_x(y) = P(y|x)$ over $\mathcal{Y}$ for each $x \in \mathcal{X}$. The adaptation of the basic model to this more general situation offers no difficulty and can be found in many texts, such as [78, 212, 200].

Another idea that has been very productive is that of producing good predictors from weak predictors (that is, they are correct only slightly better than a random predictor). The generation of weak predictors is relatively easy, and it is a remarkable fact that there is a simple and efficient algorithm to build a strong predictor from weak ones. They are known as *boosting algorithms*, and the most popular, AdaBoost (adaptive boosting), was introduced in [64], a work in which a bound on the empirical error of the algorithm is established, an analysis is made in terms of the VC dimension, and applications to problems of both multiple classification (such as face recognition) and regression are indicated. You can find an excellent exposition of it in [176, chapter 10].

Regarding neurons and neural networks, we have adopted the scheme of [204]. There are more detailed versions in many texts, such as [5, 80, 140]. We also mention the panoramic article [174].

Finally, [49] counts as the first article in which the capacity of neural networks to approximate any continuous function uniformly over compact sets was established. See also [91] and [90]. The article [220] offers an updated assessment of this problem.

B.3 Notes to section 3 (Approximation)

Sobolev spaces. A function $f : \Omega \to \mathbb{R}$ is of the Sobolev class $\mathcal{H}^{s,p}(\Omega)$ if f and its derivatives up to order s are integrable in $L^p(\Omega)$; see [170] for more details.

Asymptotic notations. Given functions $f, g : \mathbb{R} \to \mathbb{R}_+$, we say that $g(t)$ is $O(f(t))$ if there exists a constant $c > 0$ such that $g(t) \leqslant cf(t)$ for $t \gg 0$. Similarly, $g(t)$ is $\Omega(f(t))$ if there exists a constant $c' > 0$ such that $g(t) \geqslant c'f(t)$ for $t \gg 0$. Finally, $g(t)$ is said to be $\Theta(f(t))$ if it is both $O(f(t))$ and $\Omega(f(t))$.

B.4 Notes to section 4 (Optimization)

Displacement convexity. The generalization of Euclidean convexity to Wasserstein gradient flows [205, chapter 23].

Wasserstein W_2 distance. Given two probability measures μ and μ' in a metric space M, $W_2(\mu, \mu')^2 = \inf_{\gamma \in \Gamma(\mu,\mu')} \int d(x, x')^2 d\gamma(x, x')$, where $\Gamma(\mu, \mu')$ is the set of probability densities on $M \times M$ with marginal densities μ and μ'. See [205, chapter 6] and, in particular, the discussion in the bibliographical notes on terminology in which it concludes that the most appropriate would be *minimum L^p metric*, or *Kantorovich metric* if it had to carry a name. However, Villani also uses the habitual terminology.

β-regular functions. A function f of class C^1 is *β-regular* if it satisfies $||\nabla f(x) - \nabla f(y)|| \leqslant \beta||x - y||$, that is, if ∇f is β-Lipschitz.

Approximate local minimum. If f is C^2 and $\nabla^2 f$ is ρ-Lipschitz, an approximate local minimum to within ϵ is a point x such that $||\nabla f(x)|| \leqslant \epsilon$ and $\nabla^2 f(x) + \sqrt{\rho\epsilon} I \prec 0$.

B.5 Notes to section 5 (Generalization)

In [158, page 18], the author finds that the image of "shattering into small pieces" induced by the notion of a "class of functions $\mathcal{F}$ *shattering* a set S" to be a bit inappropriate, and that an image evoking the "faculty of $\mathcal{F}$ to select any subset of S" would be better –but accepts it because "at least it is vivid". For the VC capacity of the function space calculated by a NN, see [17]. The bound (32) of $\mathsf{Rad}(A)$ is known as *Massart's lemma*.

Theorem 8 can be proved using McDiarmid's inequality (see section A.3). For details and more information, see [176, chapter 26], [105, 107, 106].

B.6 Notes to section 6 (Other models and open problems)

Bayesian Inference. We record some of the major texts appeared during this century: [136] (a textbook on Bayesian networks); [104] (probabilistic graphical models); [51] (modeling and reasoning with Bayesian networks); [50] (use of matrix factorization to learn Markov models; a good sample of application of singular value decomposition); [13] (Bayesian reasoning and ML); [128] (graphical models for processing incomplete data); [84] (Markov blankets in the brain, i.e., statistical boundaries that govern the interactions between events on one side and the other of the blanket).

Reinforcement learning and some applications. [190] (an extensive manual on reinforcement learning and on various related topics); [178] (a short article on ML in self-reinforcement mode for games like chess or Go); [32] (and for poker); [175] (a recent text on the state of the art in reinforcement learning); [47] (reinforcement learning in the field of robotics); [40] (reinforcement learning applied to the protein folding problem).

Pattern Theory. The father of current pattern theory can be considered to be Ulf Grenander (1923–2016). The relationship of this theory with ML is profound, and it seems clear that all its potential has not been explored in this direction. Certainly, one can cite the *Lectures in Pattern Theory* by Grenander (Springer LNiM 18, 24, 33), but the version that seems most suitable as inspiration for ML is David Mumford and Agnes Desolneux's approach in volume [131] and the monograph [221].

ML and intelligence. In general lines, there are two large branches under construction focused on the study of learning and intelligence from the algorithmic point of view. One is represented by ML, with all its variants, of which we have attempted to provide a panoramic presentation in this work. The purpose of the other branch, on the other hand, is to find out the structure and functioning of the brains of the animal kingdom, mainly those of mammals and, very especially, those of humans. These two branches are still far from converging, but everything seems to indicate that they will do so at some future time. Upon finishing this article, [79] has been published, a remarkable book both for its popular style and for the references to the essential research works on which its arguments are based. We strongly recommend it to everyone who has an interest in the evolution of the interaction between the two branches, an interdisciplinary crossroads in which we believe many of the techniques we have described up to here will play a prominent role, perhaps with modifications and perhaps with others not yet invented.

Acknowledgments

This English version, produced in April 2026, benefited from the assistance of Gemini in converting the original Catalan manuscript from PDF into a LaTeX format.

For the original Catalan edition, the authors would like to thank Joaquim Bruna i Floris for his catalytic role in establishing this collaboration; Manuel Udina for his precise observations; and the anonymous reviewers and editors for their insightful suggestions and detailed reports.

Written largely during the COVID-19 pandemic, this work was made possible by collaborative platforms such as Overleaf, local editors like TeXstudio, and communication tools including Skype and Zoom.

References

[1] I. Aizenberg, *Complex-Valued Neural Networks with Multi-Valued Neurons*. No. 353 in Studies in Computational Intelligence, Springer-Verlag, 2011.

[2] L. Alessandretti, A. Baronchelli, and Y.-H. He, "Machine learning meets number theory: The data science of Birch-Swinnerton-Dyer," 2019. arXiv pdf.

[3] E. Alpaydin, *Introduction to Machine Learning* (4th edition). Adaptive Computation and Machine Learning, MIT Press, 2020.

[4] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[5] M. A. Arbib, ed., *The Handbook of Brain Theory and Neural Networks* (2nd edition). MIT Press, 2003.

[6] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," 2019. arXiv pdf.

[7] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[8] V. Arya, R. K. E. Bellamy, P.-Y. Chen, and seventeen others, "One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques," 2019. arXiv pdf.

[9] S. Ayesha, M. Hanif, and R. Talib, "Overview and comparative study of dimensionality reduction techniques for high dimensional data," *Information Fusion*, vol. 59, pp. 44–58, 2020.

[10] F. Bach, "Breaking the curse of dimensionality with convex neutral networks," *Journal of Machine Learning Research*, vol. 18, no. 19, pp. 1–53, 2017.

[11] V. E. Balas, S. S. Roy, D. Sharma, and P. Samui, eds., *Handbook of Deep Learning Applications*. No. 136 in Smart Innovation, Systems and Technologies, Springer Nature Switzerland AG, 2019.

[12] M.-F. Balcan, "Rademacher complexity." Course CS 8803: Machine learning theory, Georgia Tech, 2011. Lecture Notes.

[13] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[14] E. Bareinboim and J. Pearl, "Causal inference and the data-fusion problem," *Proceedings National Academy of Sciences* (USA), vol. 113, no. 27, pp. 7345–7352, 2016.

[15] A. R. Barron, "Approximation and estimation bounds for artificial neural networks," *Machine Learning*, vol. 14, no. 1, pp. 115–133, 1994.

[16] P. L. Bartlett, O. Bousquet, and S. Mendelson, "Local Rademacher complexities," *Annals of Statistics*, vol. 33, no. 4, pp. 1497–1537, 2005.

[17] P. L. Bartlett and W. Maass, "Vapnik-Chervonenkis dimension of neural nets," in *The Handbook of Brain Theory and Neural Networks* (M. Arbib, ed.), pp. 1188–1192, The MIT Press, 2003.

[18] E. Bayro-Corrochano, *Geometric Computing: for Wavelet Transforms, Robot Vision, Learning, Control and Action*. Springer, 2010.

[19] E. Bayro-Corrochano, *Geometric Algebra Applications Vol. I: Computer Vision, Graphics and Neurocomputing*. Springer International Publishing, 2019.

[20] E. J. Bayro-Corrochano and N. Arana-Daniel, "Clifford support vector machines for classification, regression, and recurrence," *IEEE Transactions on Neural Networks*, vol. 21, no. 11, pp. 1731–1746, 2010.

[21] E. Bayro-Corrochano and G. Scheuermann, eds., *Geometric Algebra Computing: in Engineering and Computer Science*. Springer-Verlag, 2010.

[22] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias-variance trade-off," *Proceedings National Academy of Sciences* (USA), vol. 116, no. 32, pp. 15849–15854, 2019. arXiv pdf.

[23] M. Belkin, S. Ma, and S. Mandal, "To understand deep learning we need to understand kernel learning," 2018. arXiv pdf.

[24] A. Bietti and J. Mairal, "Group invariance, stability to deformations, and complexity of deep convolutional representations," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 876–924, 2019.

[25] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[26] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Advances in Neural Information Processing Systems 20: Proceedings of the 2007 Conference*, pp. 161–168, Neural Information Processing Systems Foundation, Inc. (NIPS), 2008.

[27] O. Bousquet and A. Elisseeff, "Stability and generalization," *Journal of Machine Learning Research*, vol. 2, no. 3, pp. 499–526, 2002.

[28] O. Bousquet, Y. Klochkov, and N. Zhivotovskiy, "Sharper bounds for uniformly stable algorithms," in *Proceedings of Machine Learning Research*, vol. 125, pp. 610–626, 2020. Proceedings of the 33rd Conference on Learning Theory, 9-12 July 2020.

[29] F. Brackx, E. Hitzer, and S. J. Sangwine, "History of quaternion and Clifford-Fourier transforms and wavelets," in *Quaternion and Clifford Fourier Transforms and Wavelets*, Trends in Mathematics, pp. xi–xxvii, Birkhäuser, 2013.

[30] M. Braverman, E. Hazan, M. Simchowitz, and B. Woodworth, "The gradient complexity of linear regression," in *Proceedings of Machine Learning Research*, vol. 125, pp. 627–647, 2020. Proceedings of the 33rd Conference on Learning Theory, 9-12 July 2020.

[31] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[32] N. Brown and T. Sandholm, "Superhuman AI for multiplayer poker," *Science*, vol. 365, no. 6456, pp. 885–890, 2019. Science link.

[33] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[34] S. Bubeck, "Convex optimization: Algorithms and complexity," 2014. arXiv pdf, v2 2015.

[35] S. Buchholz, *A theory of neural computation with Clifford algebras*. PhD thesis, Christian-Albrechts Universität Kiel, 2005.

[36] S. Buchholz and G. Sommer, "Quaternionic spinor MLP," in *ESANN'2000 Proceedings*, pp. 377–382, D-Facto, 2000. European Symposium on Artificial Neural Networks, Bruges (Belgium), 26-28 April 2000.

[37] S. Buchholz and G. Sommer, "Clifford algebra multilayer perceptrons," in *Geometric Computing with Clifford Algebras*, pp. 315–334, Springer, 2001.

[38] T. Bülow and G. Sommer, "Quaternionic Gabor filters for local structure classification," in *Proceedings. Fourteenth International Conference on Pattern Recognition* (Cat. No. 98EX170), vol. 1, pp. 808–810, IEEE, 1998.

[39] T. Bülow and G. Sommer, "Hypercomplex signals—a novel extension of the analytic signal to the multidimensional case," *IEEE Transactions on signal processing*, vol. 49, no. 11, pp. 2844–2852, 2001.

[40] E. Callaway, "'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures," *Nature*, vol. 588, no. 7837, pp. 203–204, 2020.

[41] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, "Machine learning and the physical sciences," *Reviews of Modern Physics*, vol. 91, no. 4, p. 045002, 2019. pdf.

[42] W. L. Chan, H. Choi, and R. Baraniuk, "Quaternion wavelets for image analysis and processing," in *IEEE International Conference on Image Processing (ICIP 2004)*, vol. 5, pp. 3057–3060, IEEE, 2004.

[43] A. Chen, G. M. Rotskoff, J. Bruna, and E. Vanden-Eijnden, "A dynamical central limit theorem for shallow neural networks," 2020. arXiv pdf.

[44] L. Chizat and F. Bach, "On the global convergence of gradient descent for over-parameterized models using optimal transport," 2018. arXiv pdf.

[45] L. Chizat, E. Oyallon, and F. Bach, "On lazy training in differentiable programming," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 2937–2947, Curran Associates, 2019.

[46] K. Cho, *Foundations and advances in deep learning*. PhD thesis, Aalto University School of Science, 2014.

[47] A. Colomé and C. Torras, *Reinforcement Learning of Bimanual Robot Skills*, vol. 134 of *Springer Tracts in Advanced Robotics*. Springer, 2020.

[48] M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho, "Discovering symbolic models from deep learning with inductive biases," 2020. arXiv pdf.

[49] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[50] G. Cybenko and V. Crespi, "Learning hidden Markov models using nonnegative matrix factorization," *IEEE Transactions on Information Theory*, vol. 57, no. 6, pp. 3963–3970, 2011.

[51] A. Darwiche, *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[52] A. Darwiche, "Human-level intelligence or animal-like abilities?," *Communications of the ACM*, vol. 61, no. 10, pp. 56–67, 2018.

[53] I. Diakonikolas, D. M. Kane, and N. Zarifis, "Near-optimal SQ lower bounds for agnostically learning halfspaces and ReLUs under Gaussian marginals," 2020. arXiv pdf. SQ: statistical query.

[54] C. Domingo-Enrich, S. Jelassi, A. Mensch, G. Rotskoff, and J. Bruna, "A mean-field analysis of two-player zero-sum games," 2020. arXiv pdf.

[55] D. L. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," 2000. AMS Math Challenges Lecture 2000, 33 pages, pdf in sunju.org.

[56] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 1675–1685, PMLR, 2019. PMLR pdf.

[57] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," in *7th International Conference on Learning Representations*, OpenReview.net, 2019. arXiv pdf.

[58] R. Eldan and O. Shamir, "The power of depth for feedforward neural networks," in *Proceedings of Machine Learning Research*, vol. 49, pp. 907–940, 2016. PMLR pdf.

[59] H. J. Escalante, S. Escalera, I. Guyon, X. Baró, Y. Güçlütürk, U. Güçlü, and M. van Gerven, eds., *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Challenges in Machine Learning, Springer, 2018.

[60] C. Fang, J. D. Lee, P. Yang, and T. Zhang, "Modeling from features: a mean-field framework for over-parameterized deep neural networks," 2020. arXiv pdf.

[61] M. Felsberg and G. Sommer, "The monogenic signal," *IEEE Transactions on signal processing*, vol. 49, no. 2, pp. 313–3144, 2001. Uni-Kiel pdf.

[62] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[63] S. Franchini, A. Gentile, F. Sorbello, G. Vassallo, and S. Vitabile, "Clifford algebra based edge detector for color images," in *2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 84–91, IEEE, 2012.

[64] Y. Freund and R. E. Schapire, "Game theory, on-line prediction and boosting," in *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pp. 325–332, IEEE, 1996.

[65] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, vol. 1, no. 2, pp. 119–130, 1988.

[66] D. Garcia-Gasulla, A. Cortés, S. Alvarez-Napagao, and U. Cortés, "Signs for ethical AI: A route towards transparency," 2020. arXiv pdf.

[67] C. J. Gaudet and A. S. Maida, "Deep quaternion networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2018.

[68] A. Gelman, "Causality and statistical learning," 2011. arXiv pdf.

[69] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, 2015.

[70] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[71] S. Goel, A. Gollakota, Z. Jin, S. Karmalkar, and A. Klivans, "Superpolynomial lower bounds for learning one-layer neural networks using gradient descent," 2020. arXiv pdf.

[72] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Adaptive Computation and Machine Learning, MIT Press, 2016.

[73] L. Greengard, *The rapid evaluation of potential fields in particle systems*. ACM Distinguished Dissertations, MIT Press, 1988.

[74] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," 2020. arXiv pdf.

[75] N. Guberman, "On complex valued convolutional neural networks," 2016. MSc Thesis, School of Computer Science and Engineering, The Hebrew University of Jerusalem, arXiv pdf.

[76] J. Guttag, *Introduction to computation and programming using Python: With application to understanding data*. MIT Press, 2016.

[77] S. Hanneke and S. Kpotufe, "A no-free-lunch theorem for multitask learning," *The Annals of Statistics*, vol. 50, no. 6, pp. 3119–3143, 2022. AoS pdf.

[78] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction* (2nd edition). Springer Series in Statistics, Springer, 2009.

[79] J. Hawkins, *A Thousand Brains: A New Theory of Intelligence*. Basic Books, 2021. With a foreword by Richard Dawkins.

[80] S. Haykin, *Neural networks and learning machines*. Pearson, 2009.

[81] Y.-H. He, "The Calabi-Yau Landscape: from Geometry, to Physics, to Machine-Learning," 2018. arXiv pdf.

[82] Y.-H. He and M. Kim, "Learning algebraic structures: preliminary investigations," 2019. arXiv pdf.

[83] Y.-H. He and S.-T. Yau, "Graph Laplacians, Riemannian manifolds and their machine-learning," 2020. arXiv pdf.

[84] I. Hipolito, M. J. D. Ramstead, L. Convertino, A. Bhat, K. Friston, and T. Parr, "Markov blankets in the brain," 2021. ScienceDirect.

[85] A. Hirose (editor), *Complex-valued neural networks: Theories and applications*, vol. 5 of *Series on Innovative Intelligence*. World Scientific, 2003.

[86] A. Hirose, *Complex-valued neural networks* (second edition). Studies in Computational Intelligence, Springer, 2012. Japanese edition 2004, first English edition 2006.

[87] A. Hirose (editor), *Complex-valued neural networks: Advances and applications*. IEEE Computational Intelligence, John Wiley & Sons, 2013.

[88] E. Hitzer and S. J. Sangwine, *Quaternion and Clifford Fourier transforms and wavelets*. Trends in Mathematics, Birkhäuser, 2013.

[89] J. Hoffmann, S. Schmitt, S. Osindero, K. Simonyan, and E. Elsen, "AlgebraNets," 2020. arXiv pdf.

[90] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991. NNs pdf.

[91] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[92] T. Isokawa, N. Matsui, and H. Nishimura, "Quaternionic neural networks: Fundamental properties and applications," in *Complex-valued neural networks: Utilizing high-dimensional parameters*, pp. 411–439, IGI Global, 2009.

[93] T. Isokawa, H. Nishimura, and N. Matsui, "Quaternionic multilayer perceptron with local analyticity," *Information*, vol. 3, no. 4, pp. 756–770, 2012. MDPI.

[94] R. Iten, T. Metger, H. Wilming, L. Del Rio, and R. Renner, "Discovering physical concepts with neural networks," *Physical Review Letters*, vol. 124, no. 1, p. 010508, 2020.

[95] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Advances in neural information processing systems* (S. Bengio *et al.*, eds.), vol. 31, pp. 8571–8580, 2018. Neurips pdf.

[96] A. K. Jain, Y. Zhong, and S. Lakshmanan, "Object matching using deformable templates," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 3, pp. 267–278, 1996.

[97] C. Jin, P. Netrapalli, R. Ge, S. M. Kakade, and M. I. Jordan, "On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points," 2019. ACM pdf.

[98] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in Neural Information Processing Systems*, vol. 26, pp. 315–323, 2013. Neurips pdf.

[99] K. Kanatani, "Overviews of optimization techniques for geometric estimation," *Memoirs of the Faculty of Engineering, Okayama University*, vol. 47, pp. 1–18, 2013.

[100] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. arXiv pdf. Adam: from Adaptive Moment estimation.

[101] K. Kleese van Dam, L. Monda, N. Peters, and T. Schenkel, "From long-distance entanglement to building a nationwide quantum internet: Report of the DOE Quantum Internet Blueprint Workshop," tech. rep., OSTI, 2020. OSTI pdf.

[102] D. C. Knill and W. Richards (editors), *Perception as Bayesian inference*. Cambridge University Press, 1996. Paper #1 is "Pattern theory: a unifying perspective", by D. Mumford.

[103] M. Kobayashi, "Rotational invariance of quaternionic Hopfield neural networks," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 11, no. 4, pp. 516–520, 2016.

[104] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. Adaptive Computation and Machine Learning, MIT Press, 2009. xxxvi+1233 p.

[105] V. Koltchinskii, "Rademacher penalties and structural risk minimization," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1902–1914, 2001.

[106] V. Koltchinskii, "Rademacher complexities and bounding the excess risk in active learning," *The Journal of Machine Learning Research*, vol. 11, pp. 2457–2485, 2010.

[107] V. Koltchinskii and D. Panchenko, "Rademacher processes and bounding the risk of function learning," in *High dimensional probability, II (Seattle, WA, 1999)*, Progress in Probabability, pp. 443–457, Springer, 2000. UWA pdf.

[108] S. Kulkarni and G. Harman, *An elementary introduction to statistical learning theory*, vol. 853 of *Wiley Series in Probability and Statistics*. John Wiley & Sons, 2011.

[109] S. Y. Kung, *Kernel methods and machine learning*. Cambridge University Press, 2014.

[110] L. Lamb, A. Garcez, M. Gori, M. Prates, P. Avelar, and M. Vardi, "Graph neural networks meet neural-symbolic computing: A survey and perspective," 2020. arXiv pdf.

[111] G. Lample and F. Charton, "Deep learning for symbolic mathematics," 2019. arXiv pdf.

[112] C. Lavor, S. Xambó-Descamps, and I. Zaplana, *A Geometric Algebra Invitation to Space-Time Physics, Robotics and Molecular Geometry*. SBMA/Springerbrief, Springer, 2018.

[113] D. C. Lay, S. R. Lay, and J. J. McDonald, *Linear algebra and its applications* (fifth edition). Pearson, 2016.

[114] S. Lazendić, H. De Bie, and A. Pižurica, "Octonion sparse representation for color and multispectral image processing," in *26th European Signal Processing Conference (EUSIPCO 2018)*, pp. 608–612, IEEE, 2018.

[115] S. Lazendic, A. Pižurica, and H. De Bie, "Hypercomplex algebras for dictionary learning," in *Early Proceedings of the AGACSE 2018 Conference*, pp. 57–64, Unicamp/IMECC, 2018. pdf.

[116] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[117] J. D. Lee, Q. Lei, N. Saunshi, and J. Zhuo, "Predicting what you already know helps: Provable self-supervised learning," 2020. arXiv pdf (v2).

[118] N. Linial, Y. Mansour, and N. Nisan, "Constant depth circuits, Fourier transform, and learnability," *Journal of the ACM*, vol. 40, no. 3, pp. 607–620, 1993.

[119] D. J. C. MacKay, *Information theory, inference and learning algorithms* (version 7.2, 4th printing). Cambridge University Press, 2005.

[120] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 2008.

[121] S. Mallat, "Group invariant scattering," *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.

[122] S. Mallat, "Understanding deep convolutional networks," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150203, 2016. pdf (16 pages).

[123] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[124] S. B. McGrayne, *The theory that would not die: How Bayes' rule cracked the Enigma code, hunted down Russian submarines, & emerged triumphant from two centuries of controversy*. Yale University Press, 2011.

[125] P. Mehta and D. J. Schwab, "An exact mapping between the variational renormalization group and deep learning," 2014. arXiv pdf.

[126] S. Mei and A. Montanari, "The generalization error of random features regression: Precise asymptotics and double descent curve," 2019. arXiv pdf.

[127] M. Minsky and S. A. Papert, *Perceptrons: An introduction to computational geometry*. MIT Press, 1988. Expanded edition of "Perceptrons", 1969.

[128] K. Mohan and J. Pearl, "Graphical models for processing missing data," 2019. arXiv pdf, v2.

[129] N. Mönning and S. Manandhar, "Evaluation of complex-valued neural networks on real-valued classification tasks," 2018. arXiv pdf.

[130] E. U. Moya-Sánchez, S. Xambó-Descamps, A. S. Pérez, S. Salazar-Colores, J. Martínez-Ortega, and U. Cortés, "A bio-inspired quaternion local phase CNN layer with contrast invariance and linear sensitivity to rotation angles," *Pattern Recognition Letters*, vol. 131, pp. 56–62, 2021.

[131] D. Mumford and A. Desolneux, *Pattern theory: The stochastic analysis of real-world signals*. Applying Mathematics, A. K. Peters, 2010.

[132] K. P. Murphy, *Machine learning: A probabilistic perspective*. MIT Press, 2012.

[133] L. Nadel (editor), *Encyclopedia of Cognitive Science*. MacMillan London, 2003. 2006: Wiley Online Library.

[134] V. Nagarajan and J. Z. Kolter, "Deterministic PAC-Bayesian generalization bounds for deep networks via generalizing noise-resilience," 2019. arXiv pdf.

[135] A. Nandy and M. Biswas, *Reinforcement Learning: With Open AI, TensorFlow and Keras using Python*. Apress, 2017.

[136] R. E. Neapolitan, *Learning Bayesian networks*, vol. 38 of *Artificial Intelligence*. Pearson Prentice Hall, 2004.

[137] A. S. Nemirovskij and D. B. Yudin, *Problem complexity and method efficiency in optimization*. Discrete Mathematics, Wiley-Interscience, 1983.

[138] Y. E. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$," *Doklady Akademii Nauk*, vol. 269, pp. 543–547, 1983. In Russian.

[139] G. Nguyen, S. Dlugolinsky, M. Bobák, V. Tran, Á. L. García, I. Heredia, P. Malík, and L. Hluchỳ, "Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 77–124, 2019.

[140] M. A. Nielsen, *Neural networks and deep learning*. Determination Press, San Francisco, CA, USA, 2015. pdf.

[141] T. Nitta, "Complex-valued neural networks," in *Encyclopedia of Artificial Intelligence*, pp. 361–366, IGI Global Scientific Publishing, 2009.

[142] T. Parcollet, M. Morchid, P.-M. Bousquet, R. Dufour, G. Linarès, and R. De Mori, "Quaternion neural networks for spoken language understanding," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, pp. 362–368, 2016. pdf.

[143] T. Parcollet, Y. Zhang, M. Morchid, C. Trabelsi, G. Linarès, R. De Mori, and Y. Bengio, "Quaternion convolutional neural networks for end-to-end automatic speech recognition," 2018. arXiv pdf.

[144] A. B. Patel, T. Nguyen, and R. G. Baraniuk, "A probabilistic theory of deep learning," 2015. arXiv pdf.

[145] J. Pearl, *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, 1988.

[146] J. Pearl, "The art and science of cause and effect," 1996. A public lecture delivered November 1996 as part of the UCLA Faculty Research Lectureship Program: UCLA pdf.

[147] J. Pearl, *Causality. Models, Reasoning, and Inference* (Second edition). Cambridge University Press, 2009. First edition 2000.

[148] J. Pearl, "Causes of effects and effects of causes," *Sociological Methods & Research*, vol. 44, no. 1, pp. 149–164, 2015.

[149] J. Pearl, "Trygve Haavelmo and the emergence of causal calculus," *Econometric Theory*, vol. 31, no. 1, pp. 152–179, 2015.

[150] J. Pearl, "Theoretical impediments to machine learning with seven sparks from the causal revolution," 2018. Keynote Talk at the WSDM'18 (Web Search and Data Mining), February 5-9, 2018, Marina Del Rey, CA, USA. (arXiv pdf).

[151] J. Pearl and D. Mackenzie, *The book of Why: The new science of cause and effect*. Basic Books, 2018.

[152] J. K. Pearson and D. L. Bisset, "Neural networks in the Clifford domain," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 3, pp. 1465–1469, IEEE, 1994.

[153] J. Peters, D. Janzing, and B. Schölkopf, *Elements of causal inference: Foundations and learning algorithms*. Adaptive Computation and Machine Learning, MIT Press, 2017.

[154] H. T. Pham and P.-M. Nguyen, "A note on the global convergence of multilayer neural networks in the mean field regime," 2020. arXiv pdf.

[155] M. T. Pham and K. Tachibana, "A conformal geometric algebra based clustering method and its applications," *Advances in Applied Clifford Algebras*, vol. 26, no. 3, pp. 1013–1032, 2016.

[156] A. Pinkus, "Density in approximation theory," 2005. arXiv pdf.

[157] Z. Piran, R. Shwartz-Ziv, and N. Tishby, "The dual information bottleneck," 2020. arXiv pdf.

[158] D. Pollard, *Convergence of Stochastic Processes*. Springer Series in Statistics, Springer Verlag, 1984.

[159] C.-A. Popa, "Complex-valued convolutional neural networks for real-valued image classification," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 816–822, IEEE, 2017.

[160] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, vol. 20, pp. 1177–1184, Curran Associates, 2008.

[161] G. Rebala, A. Ravi, and S. Churiwala, *An Introduction to Machine Learning*. Springer, 2019.

[162] J. Rivera-Rovelo and E. Bayro-Corrochano, "Medical image segmentation using a self-organizing neural network and Clifford geometric algebra," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 3538–3545, IEEE, 2006.

[163] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[164] A. Rosebrock, *Deep Learning for Computer Vision with Python: ImageNet Bundle*. PyImageSearch, 2017.

[165] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological review*, vol. 65, no. 6, pp. 386–408, 1958. pdf.

[166] F. Rosenblat, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1962.

[167] G. Rotskoff, S. Jelassi, J. Bruna, and E. Vanden-Eijnden, "Global convergence of neuron birth-death dynamics," 2019. arXiv pdf.

[168] G. M. Rotskoff and E. Vanden-Eijnden, "Trainability and accuracy of neural networks: An interacting particle system approach," 2018. arXiv pdf.

[169] N. L. Roux, M. Schmidt, and F. R. Bach, "A stochastic gradient method with an exponential convergence-rate for finite training sets," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, vol. 2, pp. 2663–2671, NIPS, 2012.

[170] W. Rudin, *Functional analysis* (second edition). Internation Series in Pure and Applied Mathematics, McGraw-Hill Inc., New York, 1991. First version 1973.

[171] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[172] S. Russell and P. Norvig, *Artificial intelligence: A modern approach* (Fourth edition). Pearson, 2020.

[173] A. Said and V. Torra, eds., *Data Science in Practice*. No. 46 in Studies in Big Data, Springer International Publishing, 2019.

[174] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015. arXiv pdf.

[175] M. Sewak, *Deep Reinforcement Learning. Frontiers of Artificial Intelligence*. Springer, 2019.

[176] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[177] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," 2017. arXiv pdf.

[178] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hasabis, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[179] N. Silver, *The signal and the noise: Why so many predictions fail–but some don't*. Penguin, 2012.

[180] J. Sirignano and K. Spiliopoulos, "Mean field analysis of deep neural networks," 2019. arXiv pdf.

[181] S. Sloman, *Causal models: How people think about the world and its alternatives*. Oxford University Press, 2005.

[182] G. Sommer (editor), *Geometric computing with Clifford algebras: Theoretical foundations and applications in computer vision and robotics*. Springer, 2001.

[183] A. Spanos, *Probability Theory and Statistical Inference: Empirical Modeling with Observational Data* (second edition). Cambridge University Press, 2019.

[184] S. Sra, S. Nowozin, and S. J. Wright (editors), *Optimization for machine learning*. Neural Information Processing Series, MIT Press, 2012.

[185] J. M. Steele, "Empirical discrepancies and subadditive processes," *The Annals of Probability*, vol. 6, no. 1, pp. 118–127, 1978. AoP pdf.

[186] L. D. Stone, *Theory of optimal search*, vol. 118 of *Mathematics in Science and Engineering*. Academic Press, 1975.

[187] C. J. Stone, "Consistent nonparametric regression," *The Annals of Statistics*, vol. 5, no. 4, pp. 595–620, 1977.

[188] P. Stone, R. Brooks, E. Brynjolfsson, R. Calo, O. Etzioni, G. Hager, J. Hirschberg, S. Kalyanakrishnan, E. Kamar, S. Kraus, K. Leyton-Brown, D. Parkes, W. Press, A. Saxenian, J. Shah, M. Tambe, and A. Teller, "Artificial intelligence and life in 2030: the one hundred year study on artificial intelligence," 2016. arXiv pdf.

[189] G. Strang, *Linear algebra and learning from data*. Wellesley-Cambridge Press, 2019.

[190] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction* (2nd edition). Adaptive Computation and Machine Learning, MIT press, 2018.

[191] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013. arXiv pdf v4, 2014.

[192] S. Theodoridis, *Machine learning: a Bayesian and optimization perspective*. Academic Press, 2015.

[193] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," 2000. arXiv pdf.

[194] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5, IEEE, 2015. arXiv pdf.

[195] C. Tosh, A. Krishnamurthy, and D. Hsu, "Contrastive learning, multi-view redundancy, and linear models," 2020. PMLR pdf.

[196] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," 2017. arXiv pdf.

[197] A. B. Tsybakov, "Pointwise and sup-norm sharp adaptive estimation of functions on the Sobolev classes," *The Annals of Statistics*, vol. 26, no. 6, pp. 2420–2469, 1998.

[198] J. Bruna, S. Chintala, Y. LeCun, S. Piantino, A. Szlam, and M. Tygert, "A mathematical motivation for complex-valued convolutional networks," 2015. arXiv pdf.

[199] G. Van den Broeck, K. Mohan, A. Choi, A. Darwiche, and J. Pearl, "Efficient algorithms for Bayesian network parameter learning from incomplete data," in *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 161–170, 2015. arXiv pdf.

[200] H. Vân Lê, "Mathematical foundations of machine learning," 2020. pdf (slides).

[201] V. N. Vapnik, *Statistical learning theory*, vol. 1 of *Adaptive and Learning Systems for Signal Processing, Communications, and Control*. John Wiley & Sons, Inc., 1998.

[202] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999. IEEE pdf.

[203] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability and Its Applications*, vol. XVI, no. 2, pp. 264–280, 1971. Translated by B. Seckler. pdf.

[204] R. Vidal, J. Bruna, R. Giryes, and S. Soatto, "Mathematics of deep learning," 2017. arXiv pdf.

[205] C. Villani, *Optimal transport: Old and new*, vol. 338 of *Grundlehren der Mathematischen Wissenschaften*. Springer, 2008. xxii+973 pp.

[206] U. von Luxburg and O. Bousquet, "Distance-based classification with Lipschitz functions," *Journal of Machine Learning Research*, vol. 5, pp. 669–695, 2004. JMLR pdf.

[207] U. von Luxburg and B. Schölkopf, "Statistical learning theory: Models, concepts, and results," in *Handbook of the History of Logic*, vol. 10, Inductive logic, pp. 651–706, Elsevier, 2011. arXiv pdf.

[208] M. J. Wainwright, *High-dimensional statistics: A non-asymptotic viewpoint*, vol. 48 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, 2019.

[209] R. Wang, Y. Shi, and W. Cao, "GA-SURF: A new speeded-up robust feature extraction algorithm for multispectral images based on geometric algebra," *Pattern Recognition Letters*, vol. 127, pp. 11–17, 2019. PRL.

[210] C. Wei, K. Shen, Y. Chen, and T. Ma, "Theoretical analysis of self-training with deep networks on unlabeled data," 2020. arXiv pdf.

[211] B. Widrow and M. E. Hoff, "Adaptive switching circuits," tech. rep., Stanford Electronics Labs, Stanford University, CA, 1960.

[212] M. M. Wolf, "Mathematical foundations of supervised learning (growing lecture notes)," 2018. pdf.

[213] J. Wu, L. Xu, F. Wu, Y. Kong, L. Senhadji, and H. Shu, "Deep octonion networks," *Neurocomputing*, vol. 397, pp. 179–181, 2020.

[214] T. Wu and M. Tegmark, "Toward an AI physicist for unsupervised learning," 2018. arXiv pdf, v2.

[215] S. Xambó-Descamps, *Real spinorial groups—a short mathematical introduction*. SpringerBriefs in Mathematics, Springer, 2018.

[216] S. Xambó-Descamps, "Geometry and physics with geometric algebra," *Geometric Mechanics*, vol. 2, no. 03, pp. 337–383, 2025.

[217] S. Xambó-Descamps and E. U. Moya-Sánchez, "Geometric calculi and automatic learning–An outline," in *Systems, Patterns and Data Engineering with Geometric Calculi* (A. Delshams and S. Xambó-Descamps, eds.), ICIAM2019 SEMA SIMAI Springer Series, Springer, 2021.

[218] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating explanations for graph neural networks," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 9240–9251, 2019. NEURIPS pdf.

[219] Y. Zhao, T. Birdal, J. E. Lenssen, E. Menegatti, L. Guibas, and F. Tombari, "Quaternion equivariant capsule networks for 3D point clouds," 2019. arXiv pdf, v3.

[220] D.-X. Zhou, "Universality of deep convolutional neural networks," *Applied and Computational Harmonic Analysis*, vol. 48, no. 2, pp. 787–794, 2020.

[221] S.-C. Zhu and D. Mumford, "A stochastic grammar of images," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 4, pp. 259–362, 2007. Web site.

[222] X. Zhu, Y. Xu, H. Xu, and C. Chen, "Quaternion convolutional neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV 2018)*, pp. 631–647, Springer, 2018.

[223] A. Zweig and J. Bruna, "A functional perspective on learning symmetric functions with neural networks," 2020. arXiv pdf.

*Courant Institute & Center for Data Science, New York University. 60 Fifth Avenue, Office 612, New York, USA. <bruna@cims.nyu.edu>, http://cims.nyu.edu/ bruna/

**Departament de Matemàtiques, UPC, i Visitant BSC. Edifici OMEGA, c/ Jordi Girona 1-3, 08034 Barcelona, Spain. <sebastia.xambo@upc.edu>, https://mat.upc.edu/en/people/sebastia.xambo/