self-supervised learning, generative modelling, or reinforcement learning) are *not* our central focus. Hence, we will not review in depth influential neural pipelines such as variational autoencoders (Kingma and Welling, 2013), generative adversarial networks (Goodfellow et al., 2014), normalising flows (Rezende and Mohamed, 2015), deep Q-networks (Mnih et al., 2015), proximal policy optimisation (Schulman et al., 2017), or deep mutual information maximisation (Hjelm et al., 2019). That being said, we believe that the principles we will focus on are of significant importance in all of these areas.

The same applies for techniques used for *optimising* or *regularising* our architectures, such as Adam (Kingma and Ba, 2014), dropout (Srivastava et al., 2014) or batch normalisation (Ioffe and Szegedy, 2015).

Further, while we have attempted to cast a reasonably wide net in order to illustrate the power of our geometric blueprint, our work does not attempt to accurately summarise the *entire* existing wealth of research on Geometric Deep Learning. Rather, we study several well-known architectures in-depth in order to demonstrate the principles and ground them in existing research, with the hope that we have left sufficient references for the reader to meaningfully apply these principles to any future geometric deep architecture they encounter or devise.

2 Learning in High Dimensions

Supervised machine learning, in its simplest formalisation, considers a set of N observations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ drawn *i.i.d.* from an underlying data distribution P defined over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are respectively the data and the label domains. The defining feature in this setup is that $\mathcal{X}$ is a *high-dimensional space*: one typically assumes $\mathcal{X} = \mathbb{R}^d$ to be a Euclidean space of large dimension d.

Let us further assume that the labels y are generated by an unknown function f, such that $y_i = f(x_i)$, and the learning problem reduces to estimating the function f using a parametrised function class $\mathcal{F} = \{f_{\boldsymbol{\theta} \in \Theta}\}$. Neural networks are a common realisation of such parametric function classes, in which case $\boldsymbol{\theta} \in \Theta$ corresponds to the network weights. In this idealised setup, there is no noise in the labels, and modern deep learning systems typically operate in the so-called *interpolating regime*, where the estimated $\tilde{f} \in \mathcal{F}$ satisfies $\tilde{f}(x_i) = f(x_i)$ for all $i = 1, \ldots, N$. The performance of a learning algorithm is measured in terms of the *expected performance* on new samples drawn from

Statistical learning theory is concerned with more refined notions of generalisation based on *concentration inequalities*; we will review some of these in future work.

P, using some *loss* $L(\cdot,\cdot)$

$$\mathcal{R}(\tilde{f}) := \mathbb{E}_P \; L(\tilde{f}(x), f(x)),$$

with the squared-loss $L(y, y') = \frac{1}{2}|y - y'|^2$ being among the most commonly used ones.

A successful learning scheme thus needs to encode the appropriate notion of regularity or *inductive bias* for f, imposed through the construction of the function class $\mathcal{F}$ and the use of *regularisation*. We briefly introduce this concept in the following section.

2.1 Inductive Bias via Function Regularity

Modern machine learning operates with large, high-quality datasets, which, together with appropriate computational resources, motivate the design of rich function classes $\mathcal{F}$ with the capacity to interpolate such large data. This mindset plays well with neural networks, since even the simplest choices of architecture yields a *dense* class of functions. The capacity to approximate almost arbitrary functions is the subject of various *Universal Approximation Theorems*; several such results were proved and popularised in the 1990s by applied mathematicians and computer scientists (see e.g. Cybenko (1989); Hornik (1991); Barron (1993); Leshno et al. (1993); Maiorov (1999); Pinkus (1999)).

A set $\mathcal{A} \subset \mathcal{X}$ is said to be *dense* in $\mathcal{X}$ if its closure

$$\mathcal{A} \cup \{\lim_{i\to\infty} a_i : a_i \in \mathcal{A}\} = \mathcal{X}.$$

This implies that any point in $\mathcal{X}$ is arbitrarily close to a point in $\mathcal{A}$. A typical Universal Approximation result shows that the class of functions represented e.g. by a two-layer perceptron, $f(\mathbf{x}) = \mathbf{c}^\top \mathrm{sign}(\mathbf{A}\mathbf{x} + \mathbf{b})$ is dense in the space of continuous functions on $\mathbb{R}^d$.

Universal Approximation, however, does not imply an *absence* of inductive bias. Given a hypothesis space $\mathcal{F}$ with universal approximation, we can define a complexity measure $c : \mathcal{F} \to \mathbb{R}_+$ and redefine our interpolation problem as

$$\tilde{f} \in \arg\min_{g\in\mathcal{F}} c(g) \quad \text{s.t.} \quad g(x_i) = f(x_i) \quad \text{for} \;\; i = 1, \ldots, N,$$

i.e., we are looking for the most regular functions within our hypothesis class. For standard function spaces, this complexity measure can be defined as a *norm*, making $\mathcal{F}$ a *Banach space* and allowing to leverage a plethora of theoretical results in functional analysis. In low dimensions, splines are a workhorse for function approximation. They can be formulated as above, with a norm capturing the classical notion of smoothness, such as the squared-norm of second-derivatives $\int_{-\infty}^{+\infty} |f''(x)|^2 \mathrm{d}x$ for cubic splines.

Informally, a norm $\|x\|$ can be regarded as a "length" of vector x. A *Banach space* is a complete vector space equipped with a norm.

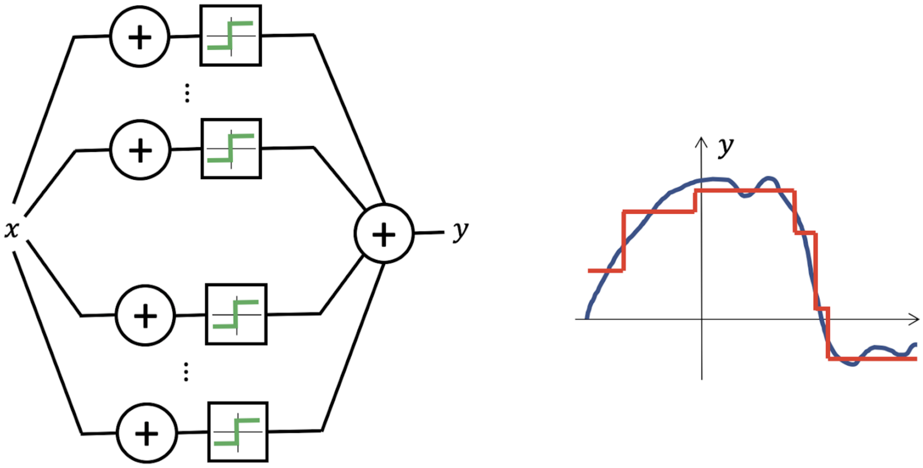

Figure 1: Multilayer Perceptrons (Rosenblatt, 1958), the simplest feed-forward neural networks, are universal approximators: with just one hidden layer, they can represent combinations of step functions, allowing to approximate any continuous function with arbitrary precision.

In the case of neural networks, the complexity measure c can be expressed in terms of the network weights, i.e. $c(f_{\boldsymbol{\theta}}) = c(\boldsymbol{\theta})$. The L_2-norm of the network weights, known as *weight decay*, or the so-called *path-norm* (Neyshabur et al., 2015) are popular choices in deep learning literature. From a Bayesian perspective, such complexity measures can also be interpreted as the negative log of the prior for the function of interest. More generally, this complexity can be enforced *explicitly* by incorporating it into the empirical loss (resulting in the so-called Structural Risk Minimisation), or *implicitly*, as a result of a certain optimisation scheme. For example, it is well-known that gradient-descent on an under-determined least-squares objective will choose interpolating solutions with minimal L_2 norm. The extension of such implicit regularisation results to modern neural networks is the subject of current studies (see e.g. Blanc et al. (2020); Shamir and Vardi (2020); Razin and Cohen (2020); Gunasekar et al. (2017)). All in all, a natural question arises: how to define effective priors that capture the expected regularities and complexities of real-world prediction tasks?

2.2 The Curse of Dimensionality

While interpolation in low-dimensions (with $d = 1, 2$ or 3) is a classic signal processing task with very precise mathematical control of estimation errors using increasingly sophisticated regularity classes (such as spline interpolants, wavelets, curvelets, or ridgelets), the situation for high-dimensional problems is entirely different.

In order to convey the essence of the idea, let us consider a classical notion of regularity that can be easily extended to high dimensions: 1-Lipschitz-functions $f : \mathcal{X} \to \mathbb{R}$, i.e. functions satisfying $|f(x) - f(x')| \leq \|x - x'\|$ for all $x, x' \in \mathcal{X}$. This hypothesis only asks the target function to be *locally* smooth, i.e., if we perturb the input x slightly (as measured by the norm $\|x - x'\|$), the output $f(x)$ is not allowed to change much. If our only knowledge of the target function f is that it is 1-Lipschitz, how many observations do we expect to require to ensure that our estimate $\tilde{f}$ will be close to f? Figure 2 reveals that the general answer is necessarily exponential in the dimension d, signaling that the Lipschitz class grows 'too quickly' as the input dimension increases: in many applications with even modest dimension d, the number of samples would be bigger than the number of atoms in the universe. The situation is not better if one replaces the Lipschitz class by a global smoothness hypothesis, such as the Sobolev Class $\mathcal{H}^s(\Omega_d)$. Indeed, classic results (Tsybakov, 2008) establish a minimax rate of approximation and learning for the Sobolev class of the order $\epsilon^{-d/s}$, showing that the extra smoothness assumptions on f only improve the statistical picture when $s \propto d$, an unrealistic assumption in practice.

A function f is in the *Sobolev class* $\mathcal{H}^s(\Omega_d)$ if $f \in L^2(\Omega_d)$ and the generalised s-th order derivative is square-integrable: $\int |\omega|^{2s+1}|\hat{f}(\omega)|^2 d\omega < \infty$, where $\hat{f}$ is the Fourier transform of f; see Section 4.2.

Fully-connected neural networks define function spaces that enable more flexible notions of regularity, obtained by considering complexity functions c on their weights. In particular, by choosing a sparsity-promoting regularisation, they have the ability to break this curse of dimensionality (Bach, 2017). However, this comes at the expense of making strong assumptions on the nature of the target function f, such as that f depends on a collection of low-dimensional projections of the input (see Figure 3). In most real-world applications (such as computer vision, speech analysis, physics, or chemistry), functions of interest tend to exhibits complex long-range correlations that cannot be expressed with low-dimensional projections (Figure 3), making this hypothesis unrealistic. It is thus necessary to define an alternative source of regularity, by exploiting the spatial structure of the physical domain and the geometric priors of f, as we describe in the next Section 3.