Locality defeats the curse of dimensionality in convolutional teacher-student scenarios

Anonymous Author(s)
Affiliation
Address
email

Abstract

Convolutional neural networks perform a local and translationally-invariant treatment of the data: quantifying which of these two aspects is central to their success remains a challenge. We study this problem within a teacher-student framework for kernel regression, using 'convolutional' kernels inspired by the neural tangent kernel of simple convolutional architectures of given filter size. Using heuristic methods from physics, we find in the ridgeless case that locality is key in determining the learning curve exponent β (that relates the test error $\epsilon_t \sim P^{-\beta}$ to the size of the training set P), whereas translational invariance is not. In particular, if the filter size of the teacher t is smaller than that of the student s, β is a function of s only and does not depend on the input dimension. We confirm our predictions on β empirically. Theoretically, in some cases (including when teacher and student are equal) it can be shown that this prediction is an upper bound on performance. We conclude by proving, using a natural universality assumption, that performing kernel regression with a ridge that decreases with the size of the training set leads to similar learning curve exponents to those we obtain in the ridgeless case.

1 Introduction

Deep Convolutional Neural Networks (CNNs) are widely recognised as the engine of the latest successes of deep learning methods, yet such a success is surprising. Indeed, any supervised learning model suffers *in principle* from the curse of dimensionality: under minimal assumptions on the function to be learnt, achieving a fixed target generalisation error ϵ requires a number of training samples P which grows exponentially with the dimensionality d of input signals [1], i.e. $\epsilon(P) \sim P^{-1/d}$. Nonetheless, empirical evidence shows that the curse of dimensionality is beaten *in practice* [2, 3, 4], with

$$\epsilon(P) \sim P^{-\beta}, \quad \beta \gg 1/d. \tag{1}$$

CNNs, in particular, achieve excellent performances on high-dimensional tasks such as image classification on ImageNet with state-of-the-art architectures, for which $\beta \approx [0.3, 0.5]$ [2]. Natural data must then possess additional structures that make them learnable. A classical idea [5] ascribes the success of recognition systems to the compositionality of data, i.e. the fact that objects are made of features, themselves made of sub-features [6, 7, 8]. In this view, the locality of CNNs plays a key role for their performance, as supported by empirical observations [9]. Yet, there is no clear analytical understanding of the relationship between compositionality of the data and learning curves.

We study this relationship in a teacher-student setting, where the true function to be learnt takes one of the following two forms:

$$f^{LC}(\boldsymbol{x}) = \sum_{i \in \mathcal{P}} g_i(\boldsymbol{x}_i), \quad f^{CN}(\boldsymbol{x}) = \sum_{i \in \mathcal{P}} g(\boldsymbol{x}_i). \tag{2}$$

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Here, $\boldsymbol{x}$ is a d-dimensional input and $\boldsymbol{x}_i$ denotes the i-th patch of $\boldsymbol{x}$, $\boldsymbol{x}_i = (x_i, \ldots, x_{i+t-1})$ where $t < d$ is the filter size of the network and i ranges in a subset $\mathcal{P}$ of $\{1, \ldots, d\}$. g_i and g are random functions, whose smoothness is controlled by some exponent α_t. Such functions can be generated for example by randomly-initialised one-hidden-layer neural networks. f^{LC} corresponds then to a *locally connected* network (LCN) [10, 11], in which the input is split into lower-dimensional patches before being processed, whereas a network enforcing invariance with respect to shifts of the input via weight sharing can be described by f^{CN}, for a suitable choice of $\mathcal{P}$. Our goal is to compute the learning curve exponent β of a student kernel performing regression on such data. The student kernel corresponds to a prior on the true function of the form described by Eq. (2), except that the filter size of the student s and its prior α_s on the smoothness of the functions g's can differ from that of the teacher. Such students include one-hidden-layer neural networks of diverging width operating in the *lazy training regime* [12, 13, 14, 15, 16].

1.1 Our contributions

We consider a teacher-student framework where the teacher is a Gaussian random field of covariance $K_T(\boldsymbol{x}, \boldsymbol{y})$ characterised by a filter size t and leading to local functions of the forms in Eq. (2), whose smoothness is characterised by an exponent $\alpha_t > 0$. Kernel regression is performed by a student kernel with $\alpha_s > 0$ and filter size s. Our main contributions are as follows:

- We use recent results based on the replica method of statistical physics on the generalisation error of kernel methods [17, 18, 19] to estimate the training curve exponent. We find that $\beta = \alpha_t/s$ if $t \leq s$ and $\alpha_t \leq 2(\alpha_s + s)$. This approach is non-rigorous, but can be proven for Gaussian fields if data are sampled on a lattice [4]. It also corresponds to a provable upper bound on β when teacher and student are equal [20].
- We confirm systematically our predictions numerically by performing ridgeless regression for various t, s and embedding dimension d.
- We use the recent framework of [21] and a natural Gaussian universality assumption to prove a lower bound on β, in the case where the ridge decreases with the size of the training set. The bound depends on s and not on d, demonstrating that the curse of dimensionality must indeed be beaten using local filters on such compositional data.

Overall, our results on the learning curve exponent β show that a locality prior allows the student to beat the curse of dimensionality on data displaying a compositional structure. By contrast, invariance for shifts can affect the prefactor entering the learning curve, but not β.

1.2 Related work

Several recent works study the role of the compositional structure of data. When such structure is hierarchical, deep convolutional networks can be much more expressive than shallow ones [6, 22, 7]. Concerning training, [23] shows that both convolutional and locally-connected networks can achieve a target generalisation error in polynomial time, whereas fully-connected networks cannot, for a class of functions which depend only on s consecutive bits of the d-dimensional input, with $s = \mathcal{O}(\log d)$. In [8] the effects of the architecture's locality are studied from a kernel perspective, using a class of deep convolutional kernels introduced in [24, 25] and characterising their Reproducing Kernel Hilbert Space (RKHS). Belonging to the RKHS ensures favourable bounds on performance and, for isotropic kernels, is a constraint on the function smoothness that becomes stringent in large d. For local functions, the corresponding constraint on smoothness is governed by the filter size s and not d [8]. Lastly, a recent work shows that weight sharing, in the absence of locality, leads to a mild improvement of the generalisation error of shift-invariant kernels [26].

By contrast, our work focuses on computing non-trivial training curve exponents in a setup where the locality and shift-invariance priors of the kernel can differ from those of the class of functions being learnt. In our setup, the latter are in general not in the RKHS of the kernel[1]. Technically, our result that the size of the student filter s controls the learning curve (and not that of the teacher t) relates to the fact that kernels are not able to detect data anisotropy (the fact that the function depends only on a subset of the coordinates) in worst case settings [28] nor in the typical case for Gaussian fields [29].

[1] A Gaussian field of covariance K is never in the RKHS of the kernel K, see e.g. [27].

2 Setup

Kernel ridge regression Kernel ridge regression is a method to learn a target function f^* from P observations $\{(\boldsymbol{x}^\mu, f^*_\mu)\}_{\mu=1}^P$, where the inputs $\boldsymbol{x}^\mu \in \mathbb{R}^d$ are i.i.d. random variables distributed according to the measure $p\left(d^d x\right)$ on $\mathbb{R}^d$, and $f^*_\mu = f^*(\boldsymbol{x}^\mu)$. Let K be a positive-definite kernel and $\mathcal{H}$ the corresponding Reproducing Kernel Hilbert Space (RKHS). The kernel rigde regression estimator f of the target function f^* is defined as

$$f = \underset{f\in\mathcal{H}}{\operatorname{argmin}} \left\{ \frac{1}{P}\sum_{\mu=1}^{P} \left(f(\boldsymbol{x}_\mu) - f^*_\mu\right)^2 + \lambda \left\|f\right\|^2_{\mathcal{H}} \right\}, \tag{3}$$

where $\|\cdot\|_{\mathcal{H}}$ denotes the RKHS norm and λ is the ridge parameter. The limit $\lambda \to 0^+$ is known as the ridgeless case and corresponds to the solution with minimum RKHS norm that interpolates the P observations. Eq. (3) is a convex optimisation problem, having the unique solution

$$f(\boldsymbol{x}) = \frac{1}{P}\sum_{\mu,\nu=1}^{P} K(\boldsymbol{x}, \boldsymbol{x}^\mu) \left(\left(\frac{1}{P}\mathbb{K}_P + \lambda \mathbb{I}_P \right)^{-1} \right)_{\mu,\nu} f^*_\nu, \tag{4}$$

where $\mathbb{K}_P$ is the *Gram matrix* defined as $(\mathbb{K}_P)_{\mu\nu} = K(\boldsymbol{x}^\mu, \boldsymbol{x}^\nu)$, and $\mathbb{I}_P$ denotes the P-dimensional identity matrix. Our goal is to compute the generalisation error, which we define as the expectation of the mean squared error over the data distribution $p\left(d^d x\right)$ and the target f^*, i.e:

$$\epsilon = \mathbb{E}_{\boldsymbol{x}, f^*}\left[\left(f(\boldsymbol{x}) - f^*(\boldsymbol{x})\right)^2 \right]. \tag{5}$$

Statistical mechanics of generalisation in kernel regression The theoretical estimation of the generalisation error is still an open problem. In [17, 18], the authors derived an approximation of ϵ by decomposing the target function in the eigenbasis of the kernel K. A positive-definite kernel K can indeed be written, by Mercer's theorem, in terms of its eigenvalues $\{\lambda_\rho\}$ and eigenfunctions $\{\phi_\rho\}$:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\rho=1}^{\infty} \lambda_\rho \phi_\rho(\boldsymbol{x}) \overline{\phi_\rho(\boldsymbol{y})}, \qquad \int p\left(d^d y\right) K(\boldsymbol{x}, \boldsymbol{y}) \phi_\rho(\boldsymbol{y}) = \lambda_\rho \phi_\rho(\boldsymbol{x}). \tag{6}$$

Let us define the kernel's features as $\psi_\rho(\boldsymbol{x}) = \sqrt{\lambda_\rho}\phi_\rho(\boldsymbol{x})$. Since the kernel's eigenfunctions form a complete basis, we can decompose the target function and estimator in terms of the features

$$f^*(\boldsymbol{x}) = \sum_\rho w^*_\rho \psi_\rho(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \sum_\rho w_\rho \psi_\rho(\boldsymbol{x}). \tag{7}$$

Using the replica method—a heuristic technique from statistical physics [30]—the following set of equations is obtained in the ridgeless limit $\lambda \to 0^+$ [17, 18]:

$$\epsilon(P) = \sum_\rho \frac{\mathbb{E}[|w^*_\rho|^2]}{\lambda_\rho} \left(\frac{1}{\lambda_\rho} + \frac{P}{t(P)} \right)^{-2} \left(1 - \frac{P\gamma(P)}{t(P)^2} \right)^{-1}, \tag{8}$$

$$t(P) = \sum_\rho \left(\frac{1}{\lambda_\rho} + \frac{P}{t(P)} \right)^{-1}, \quad \gamma(P) = \sum_\rho \left(\frac{1}{\lambda_\rho} + \frac{P}{t(P)} \right)^{-2}. \tag{9}$$

The learning curve exponent β can be extracted from the asymptotic behaviour of these expressions. In the following, we assume that the kernel and the target function have a power-law spectrum. In particular, if *i)* $\lambda_\rho = \rho^{-a}$ and *ii)* $\mathbb{E}[|c_\rho|^2] \equiv \lambda_\rho \mathbb{E}[|w^*_\rho|^2] = \rho^{-b}$, with $2a > b - 1$, then Eq. (8) and Eq. (9) yield [4, 17]

$$\epsilon(P) \sim \sum_{\rho > P} \mathbb{E}[|c_\rho|^2] \equiv \mathcal{B}(P). \tag{10}$$

Eq. (10) indicates that, given P examples, the expected generalisation error can be approximated by the tail sum of the power in the target function past the first P modes of the kernel, which we denote as $\mathcal{B}(P)$. Note that in teacher-student settings for kernel ridgeless regression [31, 32, 4, 29] additional rigorous results are available:

- if teacher and student are isotropic kernels and data are sampled on a lattice Eq. (10) can be proven rigorously [4];
- if teacher and student coincide, $\epsilon(P) \leq \mathcal{B}(P)$, i.e. this estimate is a lower bound on performance [20].

3 Kernels for local and convolutional teacher-student scenarios

In this section we introduce convolutional and local kernels and motivate our choice by considering one-hidden-layer architectures. Because of the relationship between our kernels and the Neural Tangent Kernel of one-hidden-layer convolutional neural networks, our framework encompasses regression with simple neural networks trained in the lazy training regime. For the sake of clarity we limit the discussion to inputs which are sequences in $\mathbb{R}^d$, i.e. $\boldsymbol{x} = (x_1, \ldots, x_d)$. Extension to higher-order tensorial inputs such as images $\boldsymbol{X} \in \mathbb{R}^{d \times d}$ is straightforward. To avoid dealing with the boundaries of the sequence we identify x_{i+d} with x_i for all $i = 1, \ldots, d$.

Definition 3.1 (one-hidden-layer CNN) *A one-hidden-layer convolutional network with H hidden neurons and average pooling is defined as follows,*

$$f^{CNN}(\boldsymbol{x}) = \frac{1}{\sqrt{H}} \sum_{h=1}^{H} a_h \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \sigma(\boldsymbol{w}_h \cdot \boldsymbol{x}_i), \tag{11}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the input, H is the width, σ a nonlinear activation function, $\mathcal{P} \subseteq \{1, \ldots, d\}$ is a set of patch indices and $|\mathcal{P}|$ its cardinality. For all $i \in \mathcal{P}$, $\boldsymbol{x}_i$ is an s-dimensional patch of $\boldsymbol{x}$. For all $h = 1, \ldots, H$, $\boldsymbol{w}_h \in \mathbb{R}^s$ is a filter with filter size s, $a_h \in \mathbb{R}$ is a scalar weight. The dot $\cdot$ denotes the standard Euclidean scalar product.

In the network defined above, a d-dimensional input sequence $\boldsymbol{x}$ is first mapped to s-dimensional *patches* $\boldsymbol{x}_i$, which are ordered subsequences of the input. Comparing each patch to a filter $\boldsymbol{w}_h$ and applying the activation function σ leads to a $|\mathcal{P}|$-dimensional hidden representation which is equivariant for shifts of the input. The summation over the patch index i promotes this equivariance to full invariance, leading to a model which is both local and shift-invariant as f^{CN} in Eq. (2). A model which is only local (as f^{LC} in Eq. (2)) can be obtained by lifting the constraint of weight-sharing, which forces, for each $h = 1, \ldots, H$, the same filter $\boldsymbol{w}_h$ to apply to all patches $\boldsymbol{x}_i$.

Definition 3.2 (one-hidden-layer LCN) *In the notation of (3.1), a one-hidden-layer locally-connected network with H hidden neurons is defined as follows,*

$$f^{LCN}(\boldsymbol{x}) = \frac{1}{\sqrt{H}} \sum_{h=1}^{H} \frac{1}{\sqrt{|\mathcal{P}|}} \sum_{i \in \mathcal{P}} a_{h,i} \sigma(\boldsymbol{w}_{h,i} \cdot \boldsymbol{x}_i), \tag{12}$$

For all $i \in \mathcal{P}$ and $h = 1, \ldots, H$: $\boldsymbol{x}_i$ is an s-dimensional patch of $\boldsymbol{x}$, $\boldsymbol{w}_{h,i} \in \mathbb{R}^s$ is a filter with filter size s, $a_{h,i} \in \mathbb{R}$ is a scalar weight.

Notice that the definition above reduces to that of a fully-connected network, when the filter size is set to the input dimension, $s = d$, and $\mathcal{P} = \{1\}$. With the target functions taking one of the two forms in Eq. (2), our framework contains the case where the observations are generated by neural networks such as (3.1) and (3.2). Let us now introduce the neural tangent kernels of such architectures.

Definition 3.3 (Neural Tangent Kernel) *Given a neural network function $f(\boldsymbol{x}; \boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$ denotes the complete set of parameters and N the total number of parameters, the Neural Tangent Kernel (NTK) is defined as [12]*

$$\Theta_N(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \sum_{n=1}^{N} \partial_{\theta_n} f(\boldsymbol{x}, \boldsymbol{\theta}) \partial_{\theta_n} f(\boldsymbol{y}, \boldsymbol{\theta}), \tag{13}$$

where ∂_{θ_n} denotes partial derivation w.r.t. the n-th parameter θ_n.

For one-hidden-layer networks with random, $\mathcal{O}(1)$-variance Gaussian initialisation of all the weights, and normalisation by $\sqrt{H}$ as in (3.1) and (3.2), the NTK converges to a deterministic limit $\Theta(\boldsymbol{x}, \boldsymbol{y})$ as $N \propto H \to \infty$ [12]. Furthermore, training $f(\boldsymbol{x}, \boldsymbol{\theta}) - f(\boldsymbol{x}, \boldsymbol{\theta}_0)$, with $\boldsymbol{\theta}_0$ denoting the network parameters at initialisation, under gradient descent on the mean squared error is equivalent to performing ridgeless regression with kernel $\Theta(\boldsymbol{x}, \boldsymbol{y})$ [12]. The following lemmas relate the NTK of convolutional and local architectures acting on d-dimensional inputs to those of a fully-connected architecture acting on s-dimensional inputs. Both lemmas are proved in Appendix A.

Lemma 3.1 *Call Θ^{FC} the NTK of a fully-connected network function acting on s-dimensional inputs and Θ^{CN} the NTK of a convolutional network function (3.1) with filter size s acting on d-dimensional inputs. Then*

$$\Theta^{CN}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{|\mathcal{P}|^2}\sum_{i,j\in\mathcal{P}}\Theta^{FC}(\boldsymbol{x}_i,\boldsymbol{y}_j) \tag{14}$$

As the functions in Eq. (2), Θ^{CN} is written as a combination of a lower-dimensional constituent kernel Θ^{FC} acting on patches, and the dimensionality of the constituent kernel coincides with the filter size of the corresponding network. For inputs $\boldsymbol{x}$ of fixed norm, the constituent $\Theta^{FC}(\boldsymbol{x},\boldsymbol{y})$ of network with ReLU activations displays a cusp-like nonanalyticity at the origin [33], i.e. $\Theta^{FC}(\boldsymbol{x},\boldsymbol{y}) \sim \|\boldsymbol{x}_i - \boldsymbol{y}_j\|$ plus analytic (or smaller nonanalytic) contributions. This observation extends to local kernels, via

Lemma 3.2 *Call Θ^{LC} the NTK of a locally-connected network function (3.2) with filter size s acting on d-dimensional inputs. Then*

$$\Theta^{LC}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}\Theta^{FC}(\boldsymbol{x}_i,\boldsymbol{y}_i) \tag{15}$$

Following the general structure of Eq. (14) and Eq. (15), we introduce local (K^{LC}) and convolutional (K^{CN}) student and teacher kernels, defined as sums of lower-dimensional constituent kernels C,

$$K^{CN}(\boldsymbol{x},\boldsymbol{y}) = |\mathcal{P}|^{-2}\sum_{i,j\in\mathcal{P}} C(\boldsymbol{x}_i,\boldsymbol{y}_j), \tag{16a}$$

$$K^{LC}(\boldsymbol{x},\boldsymbol{y}) = |\mathcal{P}|^{-1}\sum_{i\in\mathcal{P}} C(\boldsymbol{x}_i,\boldsymbol{y}_i). \tag{16b}$$

The kernels in Eq. (16) are characterised by the dimensionality of the constituent kernel C, or *filter size* s (for the student, or t for the teacher) and the nonanalytic behaviour of C when the two arguments approach, $C(\boldsymbol{x}_i,\boldsymbol{y}_j) \sim \|\boldsymbol{x}_i - \boldsymbol{y}_j\|^{\alpha_s}$ (for the student, or α_t for the teacher) plus analytic contributions, with $\alpha_s \neq 2m$ for $m \in \mathbb{N}$. The corresponding target function f^* and an estimator f having the form displayed in Eq. (2). The α's control the smoothness of these functions, in the sense that, if $\alpha > n \in \mathbb{N}$, then the constituent kernel C is at least n times differentiable [34].

A notable example is the NTK of ReLU networks Θ^{FC}, which presents a cusp at the origin corresponding to $\alpha_s = 1$. In addition, in the $H \to \infty$ limit, a network initialised with random weights converges to a Gaussian process [35, 36, 37, 38, 39]. For networks with ReLU activations, the covariance kernel of the process has nonanalytic behaviour with $\alpha_t = 3$ [40].

3.1 Mercer's decomposition of local and convolutional kernels

We now turn to describing how the eigendecomposition of the constituent kernel C induces an eigendecomposition of convolutional and local kernels. We work under the following assumptions,

- $i)$ The constituent kernel $C(\boldsymbol{x},\boldsymbol{y})$ on $\mathbb{R}^s \times \mathbb{R}^s$ admits the following Mercer's decomposition,

$$C(\boldsymbol{x},\boldsymbol{y}) = \sum_{\rho=1}^{\infty}\lambda_\rho\phi_\rho(\boldsymbol{x})\phi_\rho(\boldsymbol{y}), \tag{17}$$

  with (ordered) eigenvalues λ_ρ and eigenfunctions ϕ_ρ such that, with $p^{(s)}(d^s x)$ denoting the s-dimensional patch measure, $\phi_1(\boldsymbol{x}) = 1\ \forall \boldsymbol{x}$ and $\int p^{(s)}(d^s x)\phi_\rho(\boldsymbol{x}) = 0$ for all $\rho > 1$;
- $ii)$ Convolutional and local kernels from Eq. (16) have *nonoverlapping* patches, i.e. d is an integer multiple of s and $\mathcal{P} = \{1 + n \times s \,|\, n = 1, \ldots, d/s\}$ with $|\mathcal{P}| = d/s$;
- $iii)$ The s-dimensional marginals on patches of the d-dimensional input measure $p^{(s)}(d^s x)$ are all identical and equal to $p^{(s)}(d^s x)$.

The part of assumption $i)$ regarding the eigenfunctions properties is satisfied, for instance, when the constituent kernel C is isotropic and data are distributed uniformly on a d-dimensional torus. The request of nonoverlapping patches in assumption $ii)$ can be relaxed at the price of further assumptions, i.e. $C(\boldsymbol{x},\boldsymbol{y}) = \mathcal{C}(\boldsymbol{x} - \boldsymbol{y})$ and data distributed uniformly on the torus, so that C is diagonalised in Fourier space (details in Appendix B).

Lemma 3.3 (Spectra of convolutional kernels) *Let K^{CN} be a convolutional kernel defined as in Eq. (16a), with a constituent kernel C satisfying assumptions $i)$, $ii)$ and $iii)$ above. Then K^{CN} admits the following Mercer's decomposition,*

$$K^{CN}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\rho=1}^{\infty} \Lambda_\rho \Phi_\rho(\boldsymbol{x}) \overline{\Phi_\rho(\boldsymbol{y})}, \tag{18}$$

with eigenvalues and eigenfunctions

$$\Lambda_1 = \lambda_1,\ \Phi_1(\boldsymbol{x}) = 1;\ \Lambda_\rho = \frac{s}{d}\lambda_\rho,\ \Phi_\rho(\boldsymbol{x}) = \sqrt{\frac{s}{d}} \sum_{i\in\mathcal{P}} \phi_\rho(\boldsymbol{x}_i)\ \textit{for}\ \rho > 1. \tag{19}$$

Lemma 3.4 (Spectra of local kernels) *Let K^{LC} be a local kernel defined as in Eq. (16b), with a constituent kernel C satisfying assumptions $i)$, $ii)$ and $iii)$ above. Then K^{LC} admits the following Mercer's decomposition,*

$$K^{LC}(\boldsymbol{x},\boldsymbol{y}) = \Lambda_1 \Phi_1(\boldsymbol{x}) + \sum_{\rho>1}^{\infty} \sum_{i\in\mathcal{P}} \Lambda_{\rho,i} \Phi_{\rho,i}(\boldsymbol{x}) \overline{\Phi_{\rho,i}(\boldsymbol{y})}, \tag{20}$$

with eigenvalues and eigenfunctions ($\forall i \in \mathcal{P}$)

$$\Lambda_1 = \lambda_1,\ \Phi_1(\boldsymbol{x}) = 1;\ \Lambda_{\rho,i} = \frac{s}{d}\lambda_\rho,\ \Phi_{\rho,i}(\boldsymbol{x}) = \phi_\rho(\boldsymbol{x}_i)\ \textit{for}\ \rho > 1. \tag{21}$$

Under assumptions $i)$, $ii)$ and $iii)$ above, lemmas 3.3 and 3.4 follow from the definitions of convolutional and local kernels and the eigendecompositions of the constituents (see Appendix B for a proof of the lemmas and generalisation to kernels with overlapping patches). In the next section we explore the consequences of these results on the asymptotics of learning curves.

4 Asymptotic learning curves for ridgeless regression

In what follows, we consider explicitly translationally-invariant constituent kernels $C(\boldsymbol{x}_i,\boldsymbol{x}_j) = \mathcal{C}(\boldsymbol{x}_i - \boldsymbol{x}_j)$ and a d-dimensional data distribution $p(d^d x)$ which is uniform on the torus, so that all lower-dimensional marginals are also uniform on lower-dimensional tori. Under these conditions, Mercer's decomposition Eq. (17) can be written in Fourier space [41], with s-dimensional plane waves $\phi^{(s)}_{\boldsymbol{k}}(\boldsymbol{x}) = e^{i\boldsymbol{k}\cdot\boldsymbol{x}}$ as eigenfunctions and the eigenvalues coinciding with the Fourier transform of $\mathcal{C}$. All the assumptions required by lemmas 3.3 and 3.4 in the previous section are satisfied in this setting. Furthermore, for kernels with filter size s (or t) and positive smoothness exponent α_s (or α_t), the eigenvalues decay with a power $-(s+\alpha_s)$ (or $-(t+\alpha_t)$) of the modulus of the wavevector $k = \sqrt{\boldsymbol{k}\cdot\boldsymbol{k}}$ [42]. In this setting, we obtain our main result:

Theorem 4.1 *Let K_T be a d-dimensional convolutional kernel with a translationally-invariant t-dimensional constituent and leading nonanalyticity at the origin controlled by the exponent $\alpha_t > 0$. Let K_S be a d-dimensional convolutional or local student kernel with a translationally-invariant s-dimensional constituent, and with a nonanalyticity at the origin controlled by the exponent $\alpha_s > 0$. If all the t-dimensional patches of the teacher are contained in at least one of the s-dimensional patches of the student[2], and data are uniformly distributed on a d-dimensional torus, the following asymptotic equivalence holds in the limit $P \to \infty$,*

$$\mathcal{B}(P) \sim P^{-\beta}, \quad \beta = \alpha_t/s.$$

Theorem 4.1, together with Eq. (10) and the additional assumption $\alpha_t \leq 2(\alpha_s + s)$, yields the following expression for the learning curves asymptotics,

$$\epsilon(P) \sim P^{-\beta}, \quad \beta = \alpha_t/s. \tag{22}$$

As β is independent of the embedding dimension d, we conclude that the curse of dimensionality is beaten when a convolutional target is learnt with a convolutional or local kernel. In fact, Eq. (22)

[2]This condition is satisfied when $s \geq t$ in the full overlapping-patches case, while requires that s is an integer multiple of t in the nonoverlapping-patches case.

indicates that there is no asymptotic advantage in using a convolutional rather than local student when learning a convolutional task, confirming the picture that locality, not weight sharing, is the main source of the convolutional architecture's performances. The prediction in Eq. (22) is confirmed empirically, as discussed in Section 5.

Theorem 4.1 is proven in Appendix C and extended to the case of a local teacher and local student in Appendix D. Here we sketch the proof for the nonoverlapping case, which begins with the calculation of the variance of the coefficients of the target function in the student basis. By indexing the coefficients with the s-dimensional wavevectors $\boldsymbol{k}$,

$$\begin{aligned}\mathbb{E}[|c_{\boldsymbol{k}}|^2] &= \int_{[0,1]^d} d^d x \Phi_{\boldsymbol{k}}(\boldsymbol{x}) \int_{[0,1]^d} d^d y \overline{\Phi_{\boldsymbol{k}}}(\boldsymbol{y}) \mathbb{E}[f^*(\boldsymbol{x}) f^*(\boldsymbol{y})] \\ &= \int_{[0,1]^d} d^d x \Phi_{\boldsymbol{k}}(\boldsymbol{x}) \int_{[0,1]^d} d^d y \overline{\Phi_{\boldsymbol{k}}}(\boldsymbol{y}) K_T(\boldsymbol{x}, \boldsymbol{y}).\end{aligned} \tag{23}$$

If the size of teacher and student coincide, $s = t$, teacher and student have the same eigenfunctions. Thus, using the eigenvalue equation Eq. (6) of the teacher yields $\mathbb{E}[|c_{\boldsymbol{k}}|^2] \sim k^{-(\alpha_t + t)} = k^{-(\alpha_t + s)}$. After ranking eigenvalues by k, with multiplicity k^{s-1} from all the wavevectors having the same modulus k, one has

$$\mathcal{B}(P) = \sum_{\{\boldsymbol{k} | k > P^{1/s}\}} k^{-(\alpha_t + s)} \sim \int_{P^{1/s}}^{\infty} dk k^{s-1} k^{-(\alpha_t + s)} \sim P^{-\frac{\alpha_t}{s}}. \tag{24}$$

When the filter size of the teacher t is lowered, some of the coefficients $\mathbb{E}[|c_{\boldsymbol{k}}|^2]$ vanish. As the target function becomes a composition of t-dimensional constituents, the only non-zero coefficients are found for $\boldsymbol{k}$'s which lie in some t-dimensional subspaces of the s-dimensional Fourier space. These subspaces correspond to the $\boldsymbol{k}$ having at most a patch of t consecutive non-vanishing components. In other words, $\mathbb{E}[|c_{\boldsymbol{k}}|^2]$ is finite only if $\boldsymbol{k}$ is effectively t-dimensional and the integral on the right-hand side of Eq. (24) becomes t-dimensional, thus

$$\mathcal{B}(P) \sim \int_{P^{1/s}}^{\infty} dk k^{t-1} k^{-(\alpha_t + t)} \sim P^{-\frac{\alpha_t}{s}}. \tag{25}$$

If the teacher patches are not contained in the student ones, the target cannot be represented with a combination of student eigenfunctions, hence the error asymptotes to a finite value when $P \to \infty$.

5 Empirical learning curves for ridgeless regression

This section investigates numerically the asymptotic behaviour of the learning curves for our teacher-student framework. We consider different combinations of convolutional and local teachers and students with overlapping patches and Laplacian constituent kernels, i.e. $\mathcal{C}(\boldsymbol{x}_i - \boldsymbol{x}_j) = e^{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}$. In order to test the robustness of our results to the data distribution, data are uniformly generated in the hypercube $[0,1]^d$ (results in Fig. 1) or on a d-hypersphere (results in Appendix F, together with further details on the experiments). Fig. 1 shows learning curves for both convolutional (left panels) and local (right panels) students learning a convolutional target function. The results in the case of a local teacher are presented in Appendix F, and display no qualitative differences.

In the following, we always refer to Fig. 1. Panels A and B show that, under the assumptions of Theorem 4.1, with $\alpha_t = \alpha_s = 1$, our prediction $\beta = 1/s$ holds independently of the embedding dimension d. Furthermore, notice that fixing the dimension d and the teacher filter size t, the generalisation errors of a convolutional and a local student with the same filter size differ only by a multiplicative constant independent of P. Indeed, the shift-invariant nature of the convolutional student only results in a pre-asymptotic corrections to our estimate of the generalisation error $\mathcal{B}(P)$. Panels C and D show learning curves for several values of s and fixed t. The curse of dimensionality is recovered when the size of the student filters coincide with the input dimension, both for local and convolutional students. Finally, panels E and F show learning curves for fixed t and s being smaller than, equal to or larger than t. We stress that, when $s < t$ the student kernel cannot reproduce the target function, hence the error does not decrease by increasing P. All experiments are in excellent agreement with our predictions.

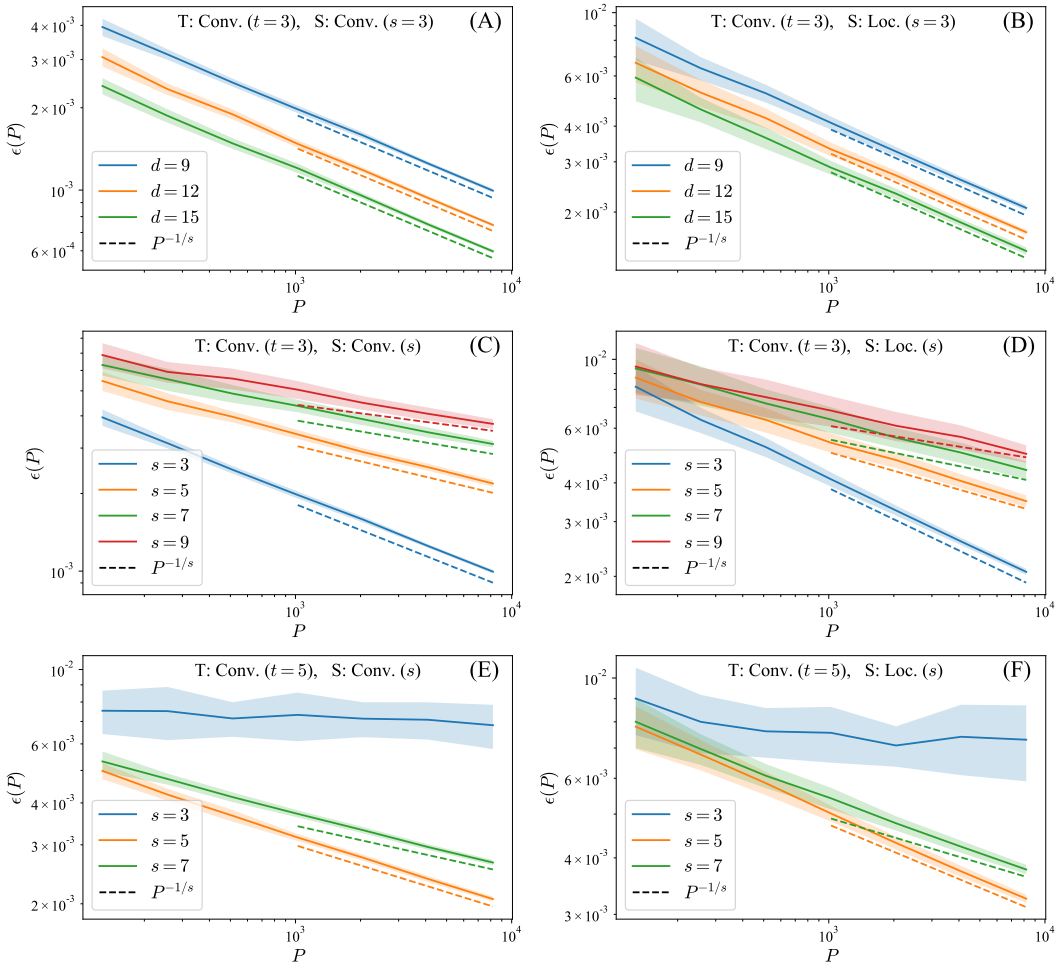

Figure 1: Learning curves for different combinations of convolutional teachers with convolutional (left panels) and local (right panels) students. The teacher and student filter sizes are denoted with t and s respectively. Data are sampled uniformly in the hypercube $[0,1]^d$, with $d=9$ if not specified otherwise. Solid lines are the results of numerical experiments averaged over 128 realizations and the shaded areas represent the empirical standard deviations. The predicted scaling are shown by dashed lines. All the panels are discussed in Section 5, while additional details on experiments are reported in Appendix F, together with additional experiments.

6 Asymptotics of learning curves with decreasing ridge

We now prove an upper bound for the exponent β implying that the curse of dimensionality is beaten by a local or convolutional kernel learning a convolutional target (as in Eq. (2)), using the framework developed in [21] and a natural assumption. We stress that this framework does not require the target function to be generated by a teacher kernel. Proofs are presented in Appendix E. Let $\mathcal{D}(\Lambda)$ denote the density of eigenvalues of the student kernel, $\mathcal{D}(\Lambda)=\sum_\rho \delta(\Lambda-\Lambda_\rho)$, with $\delta(x)$ denoting Dirac delta function. Having a random target function with coefficients c_ρ in the kernel eigenbasis having variance $\mathbb{E}[|c_\rho|^2]$, one can define the following reduced density (with respect to the teacher):

$$\mathcal{D}_T(\Lambda) = \sum_{\{\rho|\mathbb{E}[|c_\rho|^2]>0\}} \delta(\Lambda-\Lambda_\rho) \tag{26}$$

$\mathcal{D}_T(\Lambda)$ counts eigenvalues for which the target has a non-zero variance, such that:

$$\sum_\rho \mathbb{E}[|c_\rho|^2] = \int d\Lambda \mathcal{D}_T(\Lambda) c^2(\Lambda), \tag{27}$$

where the function $c(\Lambda)$ is defined by $c^2(\Lambda_\rho) = \mathbb{E}[|c_\rho|^2]$ for all ρ such that $\mathbb{E}[|c_\rho|^2] > 0$. The following theorem then follows from the results of [21].

Theorem 6.1 *Let us consider a positive-definite kernel K with eigenvalues Λ_ρ, $\sum_\rho \Lambda_\rho < \infty$, and eigenfunctions Φ_ρ learning a (random) target function f^* in kernel ridge regression (Eq. (3)) with ridge λ from P observations $f^*_\mu = f^*(\boldsymbol{x}^\mu)$, with $\boldsymbol{x}^\mu \in \mathbb{R}^d$ drawn from a certain probability distribution. Let us denote with $\mathcal{D}_T(\Lambda)$ the reduced density of kernel eigenvalues with respect to the target and $\epsilon(\lambda, P)$ the generalisation error and also assume that*

- *$i)$ For any P-tuple of indices $\rho_1, \dots, \rho_P$, the vector $(\Phi_{\rho_1}(\boldsymbol{x}^1), \dots, \Phi_{\rho_P}(\boldsymbol{x}^P))$ is a Gaussian random vector;*
- *$ii)$ The target function can be written in the kernel eigenbasis with coefficients c_ρ and $c^2(\Lambda_\rho) = \mathbb{E}[|c_\rho|^2]$, with $\mathcal{D}_T(\Lambda) \sim \Lambda^{-(1+r)}$, $c^2(\Lambda) \sim \Lambda^q$ asymptotically for small Λ and $r > 0$, $q > r$;*

Then the following equivalence holds in the joint $P \to \infty$ and $\lambda \to 0$ limit with $1/(\lambda\sqrt{P}) \to 0$:

$$\epsilon(\lambda, P) \sim \sum_{\{\rho | \Lambda_\rho < \lambda\}} \mathbb{E}[|c_\rho|^2] = \int_0^\lambda d\Lambda \mathcal{D}_T(\Lambda) c^2(\Lambda). \tag{28}$$

Note that the assumption $i)$ of the theorem on the Gaussianity of the eigenbasis does not hold in our setup where the Φ_ρ's are plane waves. However, the random variables $\Phi_\rho(\boldsymbol{x}^\mu)$ have a probability density with compact support. It is thus natural to assume that a Gaussian universality assumption holds, i.e. that Theorem 6.1 applies to our problem. With this assumption, we obtain the following

Corollary 6.1.1 *Performing kernel ridge regression in a teacher-student scenario with smoothness exponents α_t (teacher) and α_s (student), with ridge $\lambda \sim P^{-\gamma}$ and $0 < \gamma < 1/2$, under the joint hypotheses of Theorem 4.1 and Theorem 6.1, the exponent governing the asymptotic scaling of the generalisation error with P is given by:*

$$\beta = \gamma \frac{\alpha_t}{\alpha_s + s}, \tag{29}$$

which does not vanish in the limit $d \to \infty$. Furthermore, Eq. (29) depends on s and not on t as the prediction of Eq. (22).

7 Conclusions and future work

Our work shows that, even in large dimension d, a function can be learnt efficiently if it can be expressed as a sum of constituent functions each depending on a smaller number of variables t, by performing regression with a kernel that entails such a compositional structure with s-dimensional constituents. The learning curve exponent is then independent of d and governed by s if $s \geq t$, optimal for $s = t$ and null if $s < t$.

In the context of image classification, this result relates to the "Bag of Words" viewpoint. Consider for example two-dimensional images consisting of M features of t adjacent pixels, and that different classes correspond to distinct subsets of (possibly shared) features. If features can be located anywhere, then data lie on a $2M$-dimensional manifold. On the one hand, we expect a one-hidden-layer convolutional network with filter size $s \geq t$ to learn well with a learning curve exponent governed by s and independent of M. On the other hand, a fully-connected net would suffer from the curse of dimensionality for large M.

Our work does not consider that the compositional structure of real data is hierarchical, with large features that consist of smaller sub-features. It is intuitively clear that depth and locality taken together are well-suited for such data structure [8, 6]. Extending the present teacher-student framework to this case would offer valuable quantitative insights into the question of how many data are required to learn such tasks.

References

[1] Ulrike von Luxburg and Olivier Bousquet. Distance-based classification with lipschitz functions. *Journal of Machine Learning Research*, 5(Jun):669–695, 2004.

[2] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[4] Stefano Spigler, Mario Geiger, and Matthieu Wyart. Asymptotic learning curves of kernel methods: empirical data versus teacher–student paradigm. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124001, December 2020. Publisher: IOP Publishing.

[5] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

[6] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, 14(5):503–519, 2017.

[7] Arturo Deza, Qianli Liao, Andrzej Banburski, and Tomaso Poggio. Hierarchically compositional tasks and deep convolutional networks, 2020.

[8] Alberto Bietti. On approximation in deep convolutional networks: a kernel perspective, 2021.

[9] Behnam Neyshabur. Towards learning convolutions from scratch. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8078–8088. Curran Associates, Inc., 2020.

[10] Kunihiko Fukushima. Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, 20(3):121–136, 1975.

[11] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, 19:143–155, 1989.

[12] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 8580–8589, USA, 2018. Curran Associates Inc.

[13] Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

[14] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8572–8583. Curran Associates, Inc., 2019.

[15] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On Exact Computation with an Infinitely Wide Neural Net. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8141–8150. Curran Associates, Inc., 2019.

[16] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pages 2937–2947, 2019.

[17] Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum Dependent Learning Curves in Kernel Regression and Wide Neural Networks. In *International Conference on Machine Learning*, pages 1024–1034. PMLR, November 2020. ISSN: 2640-3498.

[18] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12(1):1–12, 2021.

[19] Bruno Loureiro, Cédric Gerbelot, Hugo Cui, Sebastian Goldt, Florent Krzakala, Marc Mézard, and Lenka Zdeborová. Capturing the learning curves of generic features maps for realistic data sets with a teacher-student model. *arXiv preprint arXiv:2102.08127*, 2021.

[20] Charles A Micchelli and Grace Wahba. Design problems for optimal surface interpolation. Technical report, WISCONSIN UNIV-MADISON DEPT OF STATISTICS, 1979.

[21] Arthur Jacot, Berfin Simsek, Francesco Spadaro, Clement Hongler, and Franck Gabriel. Kernel alignment risk estimator: Risk prediction from training data. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15568–15578. Curran Associates, Inc., 2020.

[22] Tommaso Poggio, Andrzej Banburski, and Qianli Liao. Theoretical issues in deep networks. *Proceedings of the National Academy of Sciences*, 117(48):30039–30045, 2020.

[23] Eran Malach and Shai Shalev-Shwartz. Computational separation between convolutional and fully-connected networks. In *International Conference on Learning Representations*, 2021.

[24] Julien Mairal. End-to-end kernel learning with supervised convolutional kernel networks. *arXiv preprint arXiv:1605.06265*, 2016.

[25] Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. *arXiv preprint arXiv:1905.12173*, 2019.

[26] Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Learning with invariances in random features and kernel models, 2021.

[27] Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

[28] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

[29] Jonas Paccolat, Stefano Spigler, and Matthieu Wyart. How isotropic kernels perform on simple invariants. *Machine Learning: Science and Technology*, 2(2):025020, March 2021. Publisher: IOP Publishing.

[30] Marc Mézard, Giorgio Parisi, and Miguel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987.

[31] Peter Sollich and Anason Halees. Learning curves for gaussian process regression: Approximations and bounds. *Neural Computation*, 14(6):1393–1428, Jun 2002.

[32] Peter Sollich. Gaussian process regression with mismatched models, 2001.

[33] Amnon Geifman, Abhay Yadav, Yoni Kasten, Meirav Galun, David Jacobs, and Basri Ronen. On the similarity between the laplace and neural tangent kernels. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1451–1461. Curran Associates, Inc., 2020.

[34] S. Spigler, M. Geiger, S. d'Ascoli, L. Sagun, G. Biroli, and M. Wyart. A jamming transition from under- to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47):474001, October 2019. Publisher: IOP Publishing.

[35] Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.

[36] Christopher KI Williams. Computing with infinite networks. *Advances in neural information processing systems*, pages 295–301, 1997.

[37] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[38] Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

[39] Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.

[40] Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

[41] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[42] Harold Widom. Asymptotic behavior of the eigenvalues of certain integral equations. ii. *Archive for Rational Mechanics and Analysis*, 17(3):215–229, 1964.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section 'our contributions' as well as the conclusions.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We don't foresee possible negative societal consequences of our work. On the contrary, we think that our work can positively contribute to the effort that the machine learning community is putting into understanding the current AI methods.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See supplemental material.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplemental material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]